

# TrakSYS 7.0 User Guide

© 2010 Parsec Automation Corp.



<b>WELCOME TO TRAKSYS .....</b>	<b>8</b>
<b>GETTING STARTED .....</b>	<b>8</b>
TRAKSYS COMPONENTS .....	8
<i>Supported Operating Systems.....</i>	9
MODELTrak.....	10
LOGICTrak.....	10
EVENTTrak .....	10
WEBTrak .....	10
ALERTTrak.....	10
SCRIBETrak .....	10
AUDITTrak.....	11
HISTORITrak .....	11
IMPROVETrak.....	11
SPCTrak .....	11
BATCHTrak .....	11
Database Manager.....	11
License Manager.....	12
NEW FEATURES IN TRAKSYS 7.0 .....	12
Batch Analysis.....	12
Material Management.....	12
Task Management .....	12
Job Centric Data Storage .....	12
System Templates .....	13
Web Based EVENTTrak.....	13
User/Labor Data Collection .....	13
New Chart and Gauge Web Parts.....	13
OPC Historical Data Access (HDA).....	13
TrakSYS Application Programming Interface (API) .....	13
Development Server .....	14
OPC Write-Back.....	14
Advanced Scripting Enhancement.....	14
Windows Server 2008 R2 Compatibility .....	14
Windows 7 Compatibility .....	14
SYSTEM REQUIREMENTS.....	14
Server.....	14
Web Clients .....	15
EVENTTrak Clients .....	15
DATABASE SECURITY .....	16
Logins.....	16
Users .....	16
LICENSING .....	16
TECHNICAL SUPPORT .....	16
<b>INSTALLING TRAKSYS.....</b>	<b>16</b>
MICROSOFT SQL SERVER FOR TRAKSYS.....	17
TRAKSYS SETUP DVD .....	17
TrakSYS Setup .....	17
EDB Setup.....	18
UPGRADING FROM A PREVIOUS VERSION .....	19

<b>MODELTRAK .....</b>	<b>19</b>
CONFIGURING YOUR SYSTEM .....	19
USER INTERFACE .....	19
View Menu .....	21
Tools Menu .....	21
TAG FINDER .....	25
LOGIC TREE .....	25
CONFIGURATION ENTITIES .....	25
EVENTTrak Nodes .....	25
LOGICTrak Instances .....	26
SCRIBETrak Nodes .....	27
Access Names .....	27
Tags .....	30
Tag Views .....	47
Tag History Definitions .....	47
Areas .....	48
Systems .....	49
Sub-Systems .....	56
Event Definitions .....	59
Task Definitions .....	63
Task Form Item Groups .....	64
Function Definitions .....	64
Event Categories .....	66
Event Category Groups .....	67
Event Category Schemes .....	67
Event Codes .....	67
KPI Calculations .....	68
KPI Calculation Views .....	74
KPI Count Categories .....	74
Sample Categories .....	74
Journal Categories .....	74
Step Categories .....	74
Step Category Groups .....	74
Step Category Schemes .....	74
Step Codes .....	75
Batch Quality Loss Categories .....	75
Products .....	75
Product Schemes .....	75
Product Attributes .....	75
Product Maps .....	76
Product Sets .....	76
Recipes .....	76
Materials .....	81
Schedules .....	82
Schedule Patterns .....	82
Shifts .....	83
Teams .....	83
SCRIBETrak Item Groups .....	84
SCRIBETrak Items .....	84
Subscribers .....	88
Devices .....	89
Subscriptions .....	89

<i>Users</i> .....	94
<i>Script Library</i> .....	96
<i>Custom Property Schemes</i> .....	97
<i>Custom Property Groups</i> .....	97
<i>Custom Properties</i> .....	97
CUSTOM PROPERTY VIEWS .....	98
GROUPS .....	98
TYPES .....	99
SYSTEM TEMPLATES .....	99
<i>Creating System Templates</i> .....	100
<i>Creating new Systems from Templates</i> .....	100
<i>Adding children under a System Template</i> .....	100
<i>Assigning Tags from Template Parents</i> .....	100
ADVANCED SCRIPT TAG API .....	101
<i>Global.Context</i> .....	101
<i>Global.Tags</i> .....	106
SCRIPT CLASS TEMPLATES .....	108
<i>LOGICTrak Script Class</i> .....	108
<i>System Script Class</i> .....	108
<i>Event Definition Script Class</i> .....	108
<i>Function Definition Script Class</i> .....	108
<i>Task Definition Script Class</i> .....	108
<i>KPI Calculation Script Class</i> .....	109
<b>LOGICTRAK</b> .....	<b>109</b>
LOGICTRAK SERVICE .....	109
<i>Starting and Stopping the Service</i> .....	109
<i>Configuring the Service</i> .....	109
LOGICTRAK CLIENT .....	110
<b>EVENTTRAK</b> .....	<b>112</b>
COMMON PROPERTIES .....	112
<i>Properties</i> .....	112
DBCONFIG CLASS .....	114
<i>Properties</i> .....	114
EVENT CATEGORY PICKER CONTROL .....	116
<i>Properties</i> .....	116
<i>Methods</i> .....	119
<i>Events</i> .....	124
EVENT DATA CONTROL .....	125
<i>Properties</i> .....	125
<i>Methods</i> .....	131
EVENT LIST CONTROL .....	135
<i>Properties</i> .....	135
<i>Methods</i> .....	140
<i>Events</i> .....	141
OEE DATA CONTROL .....	143
<i>Properties</i> .....	143
<i>Methods</i> .....	153
<i>Events</i> .....	155
PRODUCT PICKER CONTROL .....	156
<i>Properties</i> .....	156
<i>Methods</i> .....	160

<i>Events</i> .....	164
TAG DATA CONTROL .....	165
<i>Properties</i> .....	165
<i>Methods</i> .....	168
TAG LIST CONTROL .....	169
<i>Properties</i> .....	169
<i>Methods</i> .....	172
<i>Events</i> .....	181
TOUCHBOARD CONTROL .....	183
<i>Methods</i> .....	183
SAMPLE SUB-GROUP DATA CONTROL .....	183
<i>Properties</i> .....	183
<i>Methods</i> .....	186
TIME SPAN FORMAT .....	189
LOCALIZATION .....	190
SAMPLE APPLICATIONS .....	191
<i>Sample C# EVENTTrak Application</i> .....	191
<b>WEBTRAK KNOWLEDGE MANAGEMENT PORTAL .....</b>	<b>191</b>
SECURITY .....	191
<i>Configuring WEBTrak for Integration with Windows Security</i> .....	191
<i>Configuring WEBTrak for Integration with TrakSYS Security</i> .....	192
ADMINISTRATION .....	192
REPORTING FUNDAMENTALS .....	192
<i>Report Types</i> .....	192
<i>Report Variables</i> .....	200
<i>Report Users</i> .....	201
<i>User Identifiers</i> .....	202
<i>Parameter Types</i> .....	202
DIAGNOSTICS .....	203
<i>User Information</i> .....	204
<i>Server Information</i> .....	204
REPORT GROUPS .....	204
<i>Properties</i> .....	204
<i>Permissions</i> .....	205
<i>Variables</i> .....	205
REPORTS .....	206
<i>Design</i> .....	206
<i>Reports</i> .....	207
VARIABLES .....	266
<i>Global Variables</i> .....	266
<i>Report Variables</i> .....	267
<i>Local Variables</i> .....	267
REPORT PROGRAMMING .....	267
<i>Calling Reports by Key</i> .....	267
<i>Special Report Parameters</i> .....	267
LOCALIZATION .....	268
DASHBOARD REPORTS .....	268
<i>View vs. Edit Mode</i> .....	268
<i>Zones</i> .....	268
<i>Web Part Basics</i> .....	268
<i>Web Parts</i> .....	275
<i>Data Integration in the HTML Content Web Part</i> .....	281

<b>ALERTTRAK .....</b>	<b>282</b>
ALERTTRAK SETUP.....	282
SQL Server 2005 Notification Services.....	282
Create Notification Instance.....	282
ALERTTRAK CONFIGURATION.....	282
ALERTTRAK EXECUTION.....	282
<b>SCRIBETRAK .....</b>	<b>282</b>
CONFIGURING SCRIBETRAK.....	283
SCRIBETrak Groups and Items.....	283
Manual Systems.....	283
SCRIBETRAK APPLICATION.....	283
Main Menu.....	283
Security .....	283
Forms .....	283
<b>AUDITTRAK .....</b>	<b>284</b>
CONFIGURING .....	284
WINDOWS APPLICATIONS.....	284
WEBTRAK .....	284
EVENTTRAK.....	284
REPORTING.....	285
<b>HISTORITRAK .....</b>	<b>285</b>
CONFIGURING .....	285
DATA CAPTURE AND EXPIRATION .....	285
MANUAL CAPTURE.....	285
REPORTING.....	285
<b>IMPROVETRAK .....</b>	<b>285</b>
CONFIGURING .....	286
DATA ENTRY .....	286
REPORTING.....	286
<b>SPCTRAK .....</b>	<b>286</b>
CONFIGURING .....	286
DATA ENTRY FROM EVENTTRAK.....	286
AUTOMATIC DATA CAPTURE .....	286
REPORTING.....	286
<b>BATCHTRAK.....</b>	<b>287</b>
CONFIGURATION .....	287
Systems / Function Definitions .....	287
Materials .....	287
Product / Recipe.....	287
REPORTING.....	287
<b>DATABASE MANAGER .....</b>	<b>287</b>
NEW DATABASE .....	288
UPGRADE DATABASE.....	288
REPAIR LOGINS.....	288
CHECK REFERENTIAL INTEGRITY .....	288

ARCHIVE DATABASE .....	288
<i>Archive Database Wizard</i> .....	289
<i>Configuring Reports for Archive Data</i> .....	289
DELETE DATA .....	290
<i>Delete Data Wizard</i> .....	290
UPDATE DATA SOURCE .....	290
UPGRADE/PUBLISH REPORT FILES .....	291
CREATE AUDIT TRIGGERS .....	291
DELETE AUDIT TRIGGERS .....	291
CREATE NOTIFICATION INSTANCE .....	291
DELETE NOTIFICATION INSTANCE .....	291
RESYNC NOTIFICATION DATABASE .....	291
REGISTER REMOTE NOTIFICATION INSTANCE .....	292
<b>LICENSE MANAGER.....</b>	<b>292</b>
FILE LICENSE MODE .....	293
ACTIVATION.....	293
<b>TAG IMPORT.....</b>	<b>294</b>
TAG IMPORT FORMAT .....	294
TAG IMPORT WIZARD.....	294
<b>EVENT DEFINITION IMPORT .....</b>	<b>295</b>
EVENT DEFINITION IMPORT FORMAT .....	295
EVENT DEFINITION IMPORT WIZARD .....	295
<b>EFFICIENCY METRICS IN TRAKSYS .....</b>	<b>296</b>
OEE .....	296
<i>Calculating OEE</i> .....	296
<i>TrakSYS OEE Implementation</i> .....	297
TEEP .....	298
TAKT TIME .....	298
<b>ADVANCED TOPICS .....</b>	<b>299</b>
SUB-EVENTS.....	299
EVENT DEFINITION PRIORITY.....	299
<i>Sub-Priority</i> .....	300
SUB-SYSTEMS.....	301
FILTERING ERRATIC KPI COUNTER BEHAVIOR .....	301
KPI COUNTER FLOW .....	302
KPI UNITS .....	302
OPC WRITE-BACK.....	304
<b>OPC TROUBLESHOOTING GUIDE .....</b>	<b>304</b>
TRAKSYS SETUP OPC PREPARATION .....	304
COM SECURITY .....	305
TAG AND ACCESS NAME CONFIGURATION .....	307

# **Welcome to TrakSYS**

TrakSYS is a powerful suite of components for tracking production events and KPI in manufacturing environments. With TrakSYS, you can continually monitor and improve your process. Newly redesigned and rewritten with Microsoft .NET, TrakSYS is more extensible and more customizable than ever before. With its comprehensive toolset and next-generation reporting technologies, TrakSYS allows you to decide precisely what information you want and how you want to view it.

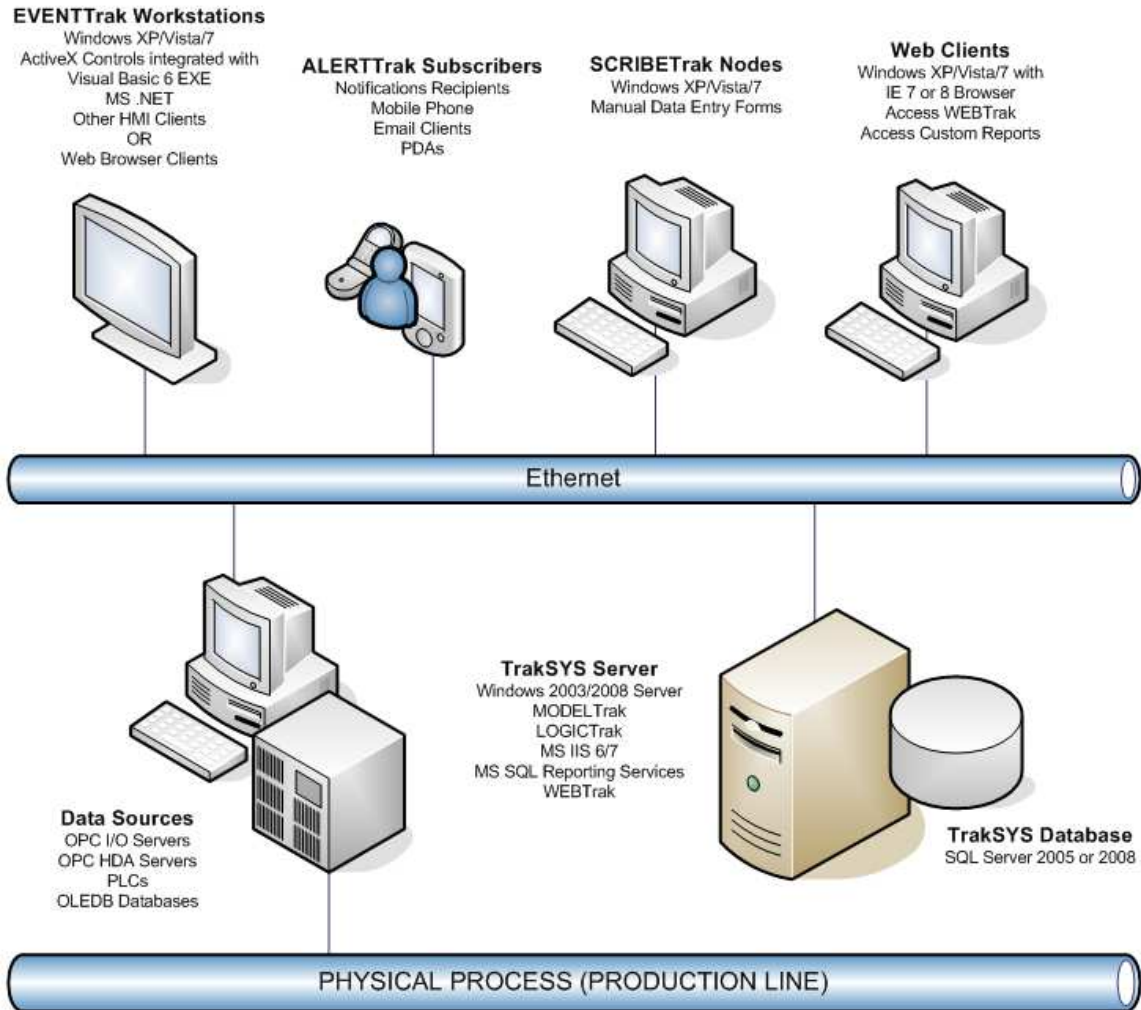
## **Getting Started**

If you are new to the TrakSYS suite of products, read through the following section to familiarize with its basic components and their functionality. Once you have completed the steps described in the section on Installing TrakSYS, you may proceed directly to Configuring Your System for guidelines on modeling your production environment in TrakSYS, or continue browsing this guide for more information on each component.

## **TrakSYS Components**

TrakSYS is a set of distributed components which allow you to configure, monitor, and administer your system. MODELTrak allows you to configure your system, LOGICTrak collects and records data, WEBTrak provides access to reports, and EVENTTrak notifies operators of recent and ongoing production Events. The diagram below illustrates the distributed nature of these components.





## TrakSYS Distributed Architecture

Note that most TrakSYS deployments will not use such a distributed configuration. For example, LOGICTrak, the Database, and WEBTrak typically reside on one physical machine. However, as your performance demands grow, you can expand these individual services to separate servers as shown above or even more servers if necessary.

## Supported Operating Systems

	Windows Clients (32 and 64 bit)			Windows Server (32 and 64 bit)		
	XP Pro	Vista	7	2003	2008	2008 R2
<b>Server Components</b>						
LOGICTrak Service	NO	NO	NO	YES	YES	YES
WEBTrak Portal	NO	NO	NO	YES	YES	YES
<b>Client Components</b>						
MODELTrak	YES	YES	YES	YES	YES	YES
LOGICTrak Client	YES	YES	YES	YES	YES	YES
EVENTTrak	YES	YES	YES	YES	YES	YES

WEBTrak Client (browser)	YES	YES	YES	YES	YES	YES
SCRIBETrak	YES	YES	YES	YES	YES	YES

## **MODELTrak**

The MODELTrak application configures the entire system, including other components like LOGICTrak and EVENTTrak. Before running TrakSYS, you must define your production environment using MODELTrak. Read the section on Configuring Your System to learn more about modeling your Lines, Equipment, Shift Schedules, Tags, and Products as a complete TrakSYS configuration.

## **LOGICTrak**

LOGICTrak is a Windows service that runs in the background collecting I/O, recording Events, tracking KPI, and sending event notifications to EVENTTrak nodes. Since LOGICTrak typically runs on a server without any users logged in, the service itself does not provide a user interface; a separate application called the LOGICTrak Client allows you to view the current status of the service.

## **EVENTTrak**

EVENTTrak is a collection of ActiveX controls which typically run within a client application written in a supported ActiveX container environment (Visual Basic, C#, InTouch, etc...). It alerts operators to recent and ongoing production events such as stoppages, rejects, and slowdowns, and allows them to acknowledge, categorize, and annotate these events according to your configuration.

## **WEBTrak**

WEBTrak Knowledge Management Portal is a comprehensive portal for TrakSYS reports and data editing tools. Once LOGICTrak has begun recording data you can use WEBTrak to view production events, calculated OEE, and other KPIs, as well as application error logs. Administrators may create and modify the reports available to users and use the editors to adjust recorded event and KPI data.

Data Manager is a set of editor user interfaces integrated into WEBTrak. These editors allow authorized administrative users to view and edit captured event and KPI data. Users can also apply post-production adjustments to KPI data and immediately view the adjusted results in the reports.

## **ALERTTrak**

ALERTTrak is the messaging and notification services for TrakSYS. Notifications can be configured in MODELTrak based on event occurrences, value changes and KPI limits. Messages can be sent to subscribers via email or SMS (text message) formats to PCs, PDAs, pagers and mobile phones. ALERTTrak is great for notifying the responsible personnel, in real time, whenever a condition requiring attention occurs.

## **SCRIBETrak**

SCRIBETrak is the manual data entry interface for TrakSYS. It is a configurable set of forms enabling the collection of data from assets where automated acquisition is not possible or cost prohibitive. SCRIBETrak allows the productivity related data (e.g. events, counts, poor quality reasons, observation, etc...) to be entered at the end of the day, shift or production run. Data collected by SCRIBETrak is stored

analyzed and visualized in the same manner as the automatically collected data (i.e., via the WEBTrak portal).

## **AUDITTrak**

AUDITTrak allows TrakSYS to be used in environments that must comply with FDA 21-CFR Part 11 requirements. AUDITTrak requires the user to enter a digital signature in the form of a user credential and password any time a change is made to the TrakSYS configuration or to data that has been collected. AUDITTrak can be configured to require one or two digital signatures when a change is made, along with a notes field for recording additional details about the change. Web-based AUDITTrak reports allow authorized users to query and view all audit information recorded to the TrakSYS database.

## **HISTORITrak**

HISTORITrak allows any Tag value within TrakSYS to be logged and recorded for trending and analysis. Tags can be configured to record their values to the database on a fixed periodic interval, when the value changes, or a combination of both. Trend reports and web parts in WEBTrak allow users to view single variables against upper and lower limits or multiple variables side by side for comparison. Single Tag trends can also be plotted versus KPI values collected by TrakSYS.

## **IMPROVETrak**

IMPROVETrak enables the tracking and analysis of actions taken to improve the productivity of manufacturing processes. Journal entries can be related to assets configured in MODELTrak and entries may be integrated into different report views in WEBTrak™ to chronicle the actions and their effects related to corrective measures, improvement scenarios, and equipment/process changes.

## **SPCTrak**

SPCTrak provides Statistical Process Control (SPC) functionality for data collected from manufacturing processes. Random sample data for user-defined process measurements and behavior can be entered through APIs available in the EVENTTrak user interface. Sample data can be collected and grouped at any stage of production. New control charts in WEBTrak automatically calculate Mean (X-Bar), Sigma (standard deviation), and Range values for sample data sets and display all necessary information in a single report for review and analysis.

## **BATCHTrak**

BATCHTrak in conjunction with other existing and new features eases the implementation of Batch performance management and detailed Electronic Batch Records (EBRs) in TrakSYS. Batch System capabilities are defined using Function Definitions and steps are specified using new Recipe definition entities in PRODUCTTrak.

## **Database Manager**

Database Manager configures connections to the TrakSYS database for the local machine. Applications such as MODELTrak, LOGICTrak, and WEBTrak rely on this information to connect to the database.

## **License Manager**

License Manager allows you to install, view, and if necessary, activate your purchased TrakSYS license features. You must use License Manager to install your license to the TrakSYS database before using the system.

## **New Features in TrakSYS 7.0**

TrakSYS 7.0 includes many new features and capabilities. The following sections describe the major changes and enhancements.

### **Batch Analysis**

Systems in TrakSYS are now process type specific. The existing System configuration has been converted to a Discrete System, and a new Batch System configuration option has been added. Within the Batch System, configurations can be created with options that are specifically tailored to a Batch type process.

### **Material Management**

In TrakSYS 7, it is possible to configure lists of Materials for use in tracking production requirements and actual usage. Simple raw Materials can be created, as well as complex mixtures which are made up of different quantities of existing raw Materials.

### **Task Management**

TrakSYS 7 contains several new features supporting the generation of pre-configured production Tasks, including the ability to define and collect inputs from the users in the production process. A few of many scenarios where the Task Management features can be used are listed below:

- Pre and post production checklists.
- Notifying responsible personnel for collecting specific data at specific intervals as part of HACCP (hazard analysis and critical control points) initiatives.
- Capturing additional production details when pre-configured conditions are present (on exception).
- Contextualizing EBRs for regulatory compliance.
- Creating and recording user-definable tasks that may not be equipment related.

### **Job Centric Data Storage**

TrakSYS 7 includes a robust infrastructure for storing Job records including planned and actual production (Job) data. The Job structures in the database become the new central hub for relating nearly all other data recorded by the software, and provide the foundation for further expansion into operations and business intelligence.

The new Job features enable the following key capabilities:

- Pre-populating the TrakSYS Job tables from external business applications that contain information about planned production quantities and target start/completion times.
- Grouping and displaying all types of TrakSYS data related to specific Jobs.
- Storing Job results like actual quantities produced and completion times for TrakSYS reporting as well as return to external business applications.

## **System Templates**

Creating and maintaining configurations with large numbers of similar Systems becomes significantly easier in TrakSYS 7 with the addition of System Template. A System Template is a specially marked System in the configuration from which other Systems can be created. Children of a System Template inherit nearly all of the parent's configuration settings as well as the sub-entities (Event/Function/Task Definitions, KPI Calculations, etc...) and their settings. Properties that must be System specific (like Tags and other select settings) are unlocked at the child System level for individual configuration. Changes made at the parent System Template are propagated to all of the associate children.

## **Web Based EVENTTrak**

New web based EVENTTrak features now allow for a true thin client plant floor interface to the server applications in TrakSYS 7. As an alternative to the existing thick client EVENTTrak ActiveX controls, several new WEBTrak Web Parts enable interaction with and feedback to the automatically recorded data from any compatible web browser.

## **User/Labor Data Collection**

To facilitate more detailed labor and personnel cost analysis, TrakSYS 7 introduces several new features that allow for more thorough tracking of resource time accrued on production assets.

## **New Chart and Gauge Web Parts**

TrakSYS 7 offers several new Web Parts in Dashboard reports specifically designed to enhance user experience, and to provide extended analytics and visualization. These reports significantly expand the scope and functionality of the WEBTrak knowledge Management Portal and are listed below:

- Production Progress Gauge
- Linear Gauges (Vertical and Horizontal)
- Pyramid Chart
- Funnel Chart
- Doughnut Chart
- Area and Line Charts
- Radar Chart
- Scatter Chart
- Histogram Chart

## **OPC Historical Data Access (HDA)**

The OPC HDA protocol defines a standard method to access historical Tag values from an existing data store (typically Historian applications). In addition to the powerful performance historian functionality built into the existing HISTORITrak component, TrakSYS 7 adds the ability to "link" to OPC HDA servers at report run time. Historical data from existing applications outside of TrakSYS can be integrated into trend reports containing native performance and operations data, without duplicating the Tag data collection.

## **TrakSYS Application Programming Interface (API)**

In TrakSYS 7 a supported and documented API is introduced, allowing application extensibility and integration with existing business applications and processes.

The TrakSYS API is made available as both a .NET Assembly that can be integrated and compiled into applications, as well as a web service based API that is served from the TrakSYS server.

## **Development Server**

For implementation environments that require a separate development environment (hardware and software), a special development license is available in TrakSYS 7 that allows configuration and data collection to be done in complete isolation from the production installation. A new database transfer tool enables the transfer of new and changed configuration from the development server to the production server.

## **OPC Write-Back**

TrakSYS 7 introduces the option for writing values back to process control devices and HMIs using the OPC communication layer. Throughout the product, opportunities to write to Tags now include the ability for those target Tags to be specified as OPC (as opposed to exclusively Virtual Tags in the past). This includes assigning values using Assign Tag logic, Product Mapping, Batch Parameter assignment, and writes using the API in Script Tags and classes.

## **Advanced Scripting Enhancement**

Organizational features have been added to the Advanced Scripting capabilities to enhance the usability and functional ease of creating scripting solutions in complex implementations.

## **Windows Server 2008 R2 Compatibility**

All TrakSYS client and server applications have been tested for compatibility and are now supported on Windows Server 2008 R2.

## **Windows 7 Compatibility**

All TrakSYS client applications have been tested for compatibility and are now supported on Windows 7 32-bit and 64-bit.

## **System Requirements**

### **Server**

TrakSYS components such as the LOGICTrak Service and WEBTrak Knowledge Management Portal make full use of a server's resources. Other running applications may impact performance as well, so the server should be as powerful as reasonably possible to maximize performance.

The TrakSYS server should have a DVD drive as the setup is distributed in DVD form only. If a DVD drive is not available on the server, the TrakSYS setup can be copied and installed from a network drive.

The following table lists server hardware requirements for different size configurations assuming all software components and database are installed.

	<b>Processor</b>	<b>Minimum RAM</b>	<b>Disk Usage per Year (estimated)</b>
--	------------------	--------------------	----------------------------------------

<b>Small Configuration</b>	Single 2 GHz	2GB	2GB
Up to 100 Event Definitions			
Up to 500 Tags			
Up to 5 Report Users			
<b>Medium Configuration</b>	Single 3 GHz	4GB	4GB
Up to 1000 Event Definitions			
Up to 5000 Tags			
Up to 25 Report Users			
<b>Large Configuration</b>	Dual 3 GHz OR Single Quad Core or Higher	8GB	8GB
Up to 5000 Event Definitions			
Up to 10000 Tags			
Up to 100 Report Users			

In general, larger configurations generate more activity, including Tag changes, Events, Intervals, report users, and etc., but the size of the configuration alone does not always provide an accurate gauge of the necessary resources. A small configuration with greater activity may produce more Tag changes and Events than a larger configuration with less activity, and therefore require more server resources. The exact server hardware requirements depend on the demands of your configuration; use the expected activity of your configuration to adjust the server resources as necessary.

## SQL Server

The TrakSYS database supports being installed on SQL Server 2005 SP3 (Standard or Enterprise) or SQL Server 2008 SP1. The reports for WEBTrak utilize and require the SQL Server Reporting Services component and ALERTTrak uses the Notification Services component which is only available in SQL Server 2005. If SQL Server 2008 is used, the ALERTTrak component of TrakSYS cannot be utilized.

## Web Clients

The minimum web client requirements are:

- ✓ Intel Pentium 2GHz processor or higher with at least 1GB of RAM.
- ✓ Microsoft Windows XP, Windows Vista, Windows 7, Windows Server 2003, or Windows Server 2008.
- ✓ Microsoft Internet Explorer 6.0 Service Pack 1 or higher.

## EVENTTrak Clients

EVENTTrak is typically run on PCs within or alongside an HMI application. Check with your ActiveX container's documentation to ensure that these workstations meet all necessary requirements. For TrakSYS, the minimum requirements are:

- ✓ Intel Pentium 2GHz processor or higher with at least 1GB of RAM.
- ✓ Microsoft Windows XP Professional, Windows Vista, Windows 7, Windows Server 2003, or Windows Server 2008.
- ✓ At least 30 MB of free hard disk space.
- ✓ Touch screen, mouse, or other pointing device.

## Database Security

TrakSYS creates a default login and user for accessing the system database, usually named EDB. The sections below describe the login and user in the default security scheme.

### Logins

The TrakSYS setup creates the following SQL Server Login:

edbApp (password Sqlapp!23)	Uses the edbApp user in the TrakSYS database.
-----------------------------	-----------------------------------------------

If the SQL Server was setup with the TrakSYS SQL Server Setup, the **sa** login password will have been set to **Sqlsa!23** (case sensitive).

### Users

edbApp	Administrative user with full ownership of the database.
--------	----------------------------------------------------------

## Licensing

TrakSYS requires a valid license in order to function. After completing the setup, you must use License Manager to install your license file before using the software. Refer to the sections Installing TrakSYS and License Manager for more information on installing TrakSYS and licensing.

## Technical Support

Parsec offers a variety of support options for troubleshooting and implementing TrakSYS. When requesting technical support, please have the following information available:

- Your software serial number.
- The version of TrakSYS you are using.
- The operating system version you are using, such as Windows Server 2003 SP2.
- The exact wording of any error messages encountered. Screenshots of the problem may be helpful as well.
- Any relevant entries in the Log Summary report, the Windows event log, or any other diagnostic log.
- Any steps you may have taken attempting to correct the problem.
- In as much detail as possible, the steps necessary to recreate the problem.
- If possible, a backup of your EDB database.

## Installing TrakSYS

TrakSYS is an extensive suite of applications and services which must be properly distributed and installed in order to function correctly. The setup aims to simplify the deployment process for administrators while still allowing them every possible option for customizing TrakSYS for a specific environment. Refer to the System Requirements section for specific hardware and software requirements prior to deploying TrakSYS.



1. Verify that your intended environment meets all requirements and determine where you will deploy the various TrakSYS components.
2. Install Microsoft SQL Server 2005 with Service Pack 2 or higher using the TrakSYS Setup DVD. SQL Server Reporting Services and Notification Services will also be installed. SQL Server will host the TrakSYS EDB database.
3. Install the desired TrakSYS components using the TrakSYS Setup on the TrakSYS Setup DVD. Depending on your environment and performance requirements, you may install all of the components on one machine or divide them across multiple clients and servers.
4. Create the TrakSYS EDB database on the database server of your choice. The main TrakSYS Setup can launch Database Manager for you automatically or you can run it later using the Database Manager shortcut.
5. Make any necessary adjustments to your database connection settings using Database Manager.
6. Run License Manager to install, and if necessary, activate, your license. You must assign specific host names to the MODELTrak, LOGICTrak, and WEBTrak feature lines.
7. Begin configuring your system using MODELTrak.

## Microsoft SQL Server for TrakSYS

The TrakSYS Setup DVD includes the installer for Microsoft SQL Server 2005 and SQL Server 2005 Service Pack 2. Selecting the "Install SQL Server 2005" option from the setup splash screen will launch the installers for SQL Server 2005 and SQL Server 2005 SP2. The installers will execute using pre-defined settings that will install the SQL Server components configured for TrakSYS. The scripted setup installs SQL Server 2005 with the following options:

- SQL Server Database Server
- SQL Server Reporting Services
- SQL Server Notification Services
- Mixed Mode Security (with Sqlsa!23 as the password for the sa account)

If other options are required, the SQL Server 2005 setup and SP3 can be launched manually from the TrakSYS Setup DVD.

## TrakSYS Setup DVD

The TrakSYS DVD includes the installation for all of the TrakSYS components and SQL Server 2005 SP3. The 64 bit version of the SQL Server installation is also provided if the installation is on a 64 bit server operating system.

## TrakSYS Setup

The main TrakSYS setup installs the base TrakSYS components and the EDB Setup. If you are installing onto a system which does not already have the Microsoft .NET 3.5 Framework SP1, the setup will install the framework automatically and then prompt for a reboot before resuming.

The TrakSYS setup provides the following component options:

MODELTrak	Installs the MODELTrak to the local machine.
LOGICTrak Client	Installs the LOGICTrak Client to the local machine.
LOGICTrak Service	Installs the LOGICTrak Service to the local machine.

WEBTrak	Installs the WEBTrak Knowledge Management Portal web site to the local machine. This option requires IIS 6 or higher.
SCRIBETrak	Installs the SCRIBETrak application and forms to the local machine.
EVENTTrak	Installs the EVENTTrak control and sample applications to the local machine.
EDB Database	Launches the secondary EDB Setup to create a new TrakSYS database on a local or remote SQL Server. You can also run this setup later using Database Manager.
Maintenance Service	Installs the Maintenance Service to the local machine
Reports	Installs a complete set of standard WEBTrak Reports to the local machine.

## EDB Setup

Database Manager and the main setup use a secondary setup called EDB Setup to create a new TrakSYS database. This setup allows you to create a new database on any accessible database server.

**Note:** The EDB Setup Wizard is not supported when running from Microsoft Vista.

The first screen prompts you to specify SQL Server login credentials. You may browse a list of available servers or directly type in the name of a server. You also have the option of using Windows or SQL Server authentication to connect to the database server. The login you specify, such as "sa", must have administrative privileges in SQL Server. Note that by default, the automated TrakSYS SQL Server setup initializes the "sa" login password to "Sqlsa!23".

Database Server	The name of the target machine running SQL Server. You may use the drop-down list to select from the list of known servers or use the Browse button to search the network for available instances of SQL Server.
Windows Authentication	Select this option to use your current Windows credentials to connect to SQL Server. You must have administrative privileges for the target SQL Server.
Server Authentication	Select this option to specify an alternative Login ID and Password to connect to SQL Server.
Login ID	The name of a SQL Server login with the necessary privileges to create the new database. The default Login ID is "sa". This value is only available when you select the SQL Server Authentication option.
Password	The password for the corresponding Login ID. This value is only available when you select the SQL Server Authentication option.

The final screen gives you the option of changing the name of the new TrakSYS database. The default name is EDB. Do not specify the name of an existing database, otherwise you may receive errors.

EDB Database Name	The name of the new TrakSYS database.
-------------------	---------------------------------------

## Upgrading From a Previous Version

When you select the EDB Database option, the main setup will offer to upgrade an existing database at the conclusion of the installation. You must specify a source database and, for major upgrades, a new destination database, as well as login credentials for each. A major upgrade from a previous version will result in an entirely new database; a minor upgrade from a previous release will directly upgrade the target database without creating a new one.

TrakSYS 7.0 includes the capability to upgrade from the previous versions. TrakSYS 7.0 setup is designed to silently uninstall the previous versions and install the 7.0 version in its place.

## MODELTrak

The MODELTrak application allows you to define the production environment you intend to monitor. You can use Areas and Systems to model physical production lines and equipment, and use Event and Function Definitions to define what production events and batch steps to track. Please read the sections below for more information on Areas, Systems, Event Definitions, Function Definitions, and other configuration entities.

## Configuring Your System

Configuring TrakSYS to properly monitor your process involves the following major tasks:

- Importing or manually creating the I/O Tags which provide real-time information from automated production lines and equipment.
- Modeling your production environment as Areas and Systems, and then adding Event Definitions based on Tags.
- Defining the Shifts, Teams, and Schedules for operations personnel, and then assigning Schedules to the appropriate Systems.
- Defining Event Categories and Event Codes used by operators at EVENTTrak workstations to classify and describe production events.
- Creating KPI Calculations using specific I/O Tags as Good, Bad, and Total product counters.
- Defining your Products and their attributes, including KPI-related information such as multipliers and theoretical rates (if applicable).
- Configuring the various software components of TrakSYS, including EVENTTrak Nodes and LOGICTrak Instances.
- Defining manual data entry forms using SCRIBETrak.
- Defining subscribers and subscriptions for notification events using ALERTTrak.
- Testing and fine-tuning the configuration to your specific needs.

## User Interface

MODELTrak presents the system configuration in a series of tree view panels which group together related entities. Due to an increased number of Configuration Entities and the increased emphasis on Tags, it is no longer feasible to arrange the entire configuration within a single tree view as in previous versions of TrakSYS. MODELTrak now organizes the configuration as follows:

## **Applications**

- LOGICTrak Instances
- EVENTTrak Nodes
- SCRIBETrak Nodes

## **Areas and Systems**

- Areas
- Systems
- Sub-Systems
- System Templates
- Event Definitions
- Function Definitions
- KPI Calculation
- System Capture Schemes
- **Error! Reference source not found.**
- KPI Capture Schemes
- System Views
- KPI Calculation Views
- Task Form Items

## **Tags**

- Tags
- Tag Views
- Access Names

## **Categories and Codes**

- Event Categories
- Event Codes
- Event Category Schemes
- KPI Count Categories
- Sample Categories
- Journal Categories
- Step Categories
- Step Codes
- Step Category Schemes
- Batch Quality Loss Categories

## **Scheduling**

- Schedules
- Shifts
- Teams

## **PRODUCTTrak**

- Product Sets
- Products
- Recipes
- Product Schemes
- Product Maps
- Materials

## **SCRIBETrak**

- SCRIBETrak Item Groups

- SCRIBETrak Items

#### **ALERTTrak**

- Subscribers
- Devices
- Subscriptions

#### **Administration**

- Users
- Script Libraries

#### **Custom Properties**

- Custom Property Schemes
- Custom Properties

### **View Menu**

#### **Show IDs**

The Show IDs menu option is used to toggle the display of configuration entity IDs in the Property dialog title bars. The configuration entity IDs are used for scripting the EVENTTrak ActiveX controls as well as custom programming against the data in the EDB database.

#### **Show Tag Detail**

The Show Tag Detail menu option is used to toggle the display of the Tag Logic Tree that appears below the list view when a Tag or Event/Function/Task Definition is selected. The Tag logic tree shows any sub-tag logic below the selected item (for example the children of a Boolean Tag or the components of a Calculation Tag).

#### **Show Banner**

The Show Banner menu option is used to display the MODELTrak banner on top of the application.

### **Tools Menu**

#### **Compile all Scripts**

This option test compiles all Script Tags and Script classes in the TrakSYS configuration. This action is useful as a final check when the automatic compile check is disabled in the TrakSYS settings.

#### **TrakSYS Settings**

The TrakSYS Settings dialog allows the configuration of many suite wide global settings. The settings are described below. Many of the settings contain default values that are applied internally if the setting is left empty.

<b>General</b>	
Resource Manager String Cache	Determines the number of string resources that are stored in memory. This helps the performance of the screen updates by reducing the number of times that translation resources need to be looked up over and over. However, increasing this value

	requires more application memory.
Decimal Symbol	In some cases, TrakSYS stores numeric float values to the database into string fields (Virtual Tag values for instance). In order to support different locals consistently, TrakSYS converts the decimal character to and from this value when storing and loading values to the database.
Auto Generate Custom Property Views	When set to True, the Custom Property Views will be automatically generated (or regenerated) each time a Custom Property, Custom Property Scheme or Custom Property Group is added, modified or deleted from the configuration.
End Active Events on LOGICTrak Start	When set to True, the LOGICTrak data collection engine will attempt to clean-up orphaned event records (those with no End Date/Time) when it starts. This setting is defaulted to False because this activity is rarely needed and can be time consuming, extending the startup process of the LOGICTrak engine unnecessarily in most cases. This setting should only be set to True at the direction of technical support.
<b>AUDITTrak</b>	
Configuration Changes Audit Mode	When set to True, TrakSYS will prompt users for digital signatures whenever modifying any configuration values (excluding PRODUCTTrak values) within MODELTrak. The signature dialogs will behave as configured by the other AUDITTrak settings below and will be presented in the format appropriate (web, Windows Form popup, custom EVENTTrak dialog, etc...).
Display Notes	When set to True, the Notes portion of the AUDITTrak digital signature dialog is displayed.
Product Changes Audit Mode	When set to True, TrakSYS will prompt users for digital signatures whenever modifying any product related configuration values within MODELTrak. The signature dialogs will behave as configured by the other AUDITTrak settings below and will be presented in the format appropriate (web, Windows Form popup, custom EVENTTrak dialog, etc...)
Require Notes	When set to True, the signee is required to enter something in the Notes portion of the AUDITTrak digital signature dialog.
Manual Data Changes Audit Mode	When set to True, TrakSYS will prompt users for digital signatures whenever modifying any data (Events, KPI Intervals, Journal Entries and SPC) after is initially captured. The signature dialogs will behave as configured by the other AUDITTrak settings below and will be presented in the format appropriate (web, Windows Form popup, custom EVENTTrak dialog, etc...).
Audit Automatic Data Changes	When set to True, TrakSYS will log all automatic data inserts and updates to the database. This setting defaults to False as it is typically not required the machine initiated data changes that are time stamped be logged. This setting should be set to True with caution as logging automatic data changes comes with a performance cost and will fill the database up quickly.
Display Signature Entry	May be set for the Primary and Secondary Signature sections. When set to True, the corresponding signature will be prompted for and required.
Signature Caption	May be set for the Primary and Secondary Signature sections.

	This is the descriptive label that will be displayed next to the signature prompt if it is enabled (above).
Retain Last User	May be set for the Primary and Secondary Signature sections. When set to True, the previous signees login will be cached (application and session specific) and be automatically filled into the signature line the next time the prompt is displayed. This affects only the login as the password cannot be set to be cached.
Authentication Type	Determines the type of authentication applied to the login(s) and password(s) in the AUDITTrak digital signature dialog. Only TrakSYS Users that are configured as Auditors will be allowed to sign for changes.
Default Domain	When the Authentication Type above is set to Windows (or Both), this is the DOMAIN that will be appended to the provided login (in an AUDITTrak digital signature dialog) if it does not contain a DOMAIN.
<b>ALERTTrak</b>	
Maximum E-Mail Failure Count	Controls the number of times the ALERTTrak notification engine will attempt to re-send failed emails before stopping.
From E-Mail Address	Specifies the from address that is used when sending ALERTTrak email messages.
Notification Services NS Control Path	Specifies the path to the NSControl.EXE which is used behind the scenes in ALERTTrak configuration. This setting should only be modified at the direction of technical support.
Base File Path	Specifies the location of the ALERTTrak XSL files. This setting should only be modified at the direction of technical support.
<b>WEBTrak</b>	
Default Page URL	Specifies the initial URL that WEBTrak will redirect to when opened to the default page. If this setting is blank, WEBTrak opens to the root of the Report tree.
Reporting Services Report Builder URL	Specifies the URL that WEBTrak uses to launch MS SQL Server Reporting Services Report Builder.
Require Web Node Registration	When this setting is checked, all nodes displaying web based EVENTTrak content will be identified by a special client cookie. The cookie is set on the client using the Web Node registration menu options in WEBTrak. These menu options are only visible when this setting is on. When this setting is unchecked, web client nodes are identified by a client IP address lookup on the network.
Use Image Pipe for Charts	When this setting is checked, all nodes displaying web based EVENTTrak content will be identified by a special client cookie. The cookie is set on the client using the Web Node registration menu options in WEBTrak. These menu options are only visible when this setting is on. When this setting is unchecked, web client nodes are identified by a client IP address lookup on the network.
Use Image Pipe for Gauges	Directs the dashboard gauges to be rendered to session memory instead of to the file system on the server. The default for this setting is off. This should only be changed if directed to do so by software support.
<b>Archive and Delete</b>	

Temporary File Path	Specifies the path used to temporarily store database backup files during an Archive operation.
Delete Temporary File	When set to True, the temporary file used for database backups during Archive operations is deleted on completion.
Temporary File Name	Specifies the filename used to temporarily store database backup files during an Archive operation.
Archive Suffix	When the Destination Database setting (below) is empty, this suffix is added to the end of the actual database name to create the new archive database name.
Destination Path	Specifies the path where the Archive database data files are created and stored.
Database Timeout	Specifies the database operation timeout (in seconds) used when Archive operations are executing.
Destination Database	Specifies the name of the Archive database that is created when the Archive action is run.
Delete Event Count	Specifies the increment in which Event data is deleted during a Delete operation. Deletes are done in these smaller chunks to avoid filling up the transaction logs.
Delete KPI Count	Specifies the increment in which KPI Interval data is deleted during a Delete operation. Deletes are done in these smaller chunks to avoid filling up the transaction logs.
Database Timeout	Specifies the database operation timeout (in seconds) used when Delete operations are executing.
<b>Log and Trace</b>	
Enabled	Determines if TrakSYS Logging is enabled. The TrakSYS Log is the primary error and application event log for all the TrakSYS applications.
Maximum File Size (bytes)	Specifies the maximum size that the Log files will grow to before being renamed and a new file started.
Log to Database	Determines if TrakSYS attempts to write errors and application events to the database Log tables.
Maximum Table Rows	Specifies the maximum number of rows maintained in the database Log tables.
Log to Windows Event Viewer	Determines if TrakSYS attempts to write errors and application events to the Windows Event Viewer.
Log Call Stack	When set to True, the Call Stack error information is recorded with the Log entry.
Log Location	When set to True, the error or event location is recorded to with Log entry.
<b>Trace Settings</b>	
Enabled	Determines if TrakSYS Tracing is enabled. The TrakSYS Trace is used for in depth debugging and should only be turned on at the direction of technical support.
Maximum Table Rows	Specifies the maximum number of rows maintained in the database Trace tables.
<b>Script</b>	
Script DLLs	This is a comma delimited file name list of external .NET assembly DLLs that are loaded and included in the LOGICTrak engine when it is run. This list should contain the name of the files only. All external DLLs must be located in the TrakSYS Program Files directory (same directory as the LOGICTrak and MODELTrak EXEs). Assemblies that are listed here may be



	referenced by Script Tags and Script Libraries.
Test Script Compile on Save	When set to True, the contents of any Script Tag or Library window will be test compiled when the dialog is closed (saved).

## Tag Finder

MODELTrak includes a versatile Tag Finder tool to help you quickly search for and select Tags. You can use any one of the following filters to narrow the results:

Name	Filters Tags by Name. Use * characters as wildcards or leave this filter blank to include all Tag Names.
Description	Filters Tags by Description. Use * characters as wildcards or leave this filter blank to include all Tag Descriptions.
Tag Group	Filters Tags by Tag Group. Select a Tag Group from the drop-down list or select [ All ] to include all Tag Groups.
Access Name	Filters Tags by Access Name. Select an Access Name from the drop-down list or select [ All ] to include all Access Names.
Type	Filters Tags by Type. Select a Tag Type from the drop-down list or select [ All ] to include all Tag Types.
Data Type	Filters Tags by Data Type. Select String, Integer, Float, or Discrete from the drop-down list or select [ All ] to include all Data Types.

## Logic Tree

The Logic Tree is a visual representation of logic that is embedded beneath a Tag or Event/Function/Task Definition. The Logic Tree is automatically displayed when an Event Definition or Tag is clicked in the main MODELTrak List View. For Event/Function/Task Definitions, the first node in the Logic Tree is the primary Tag that is referenced by the item. For Tags, the first node in the Logic Tree is the Tag itself. Double-clicking a Tag in the Logic Tree will display its Properties. If the Tag logic contains circular references, the Logic Tree will terminate rendering at the point at which the circular reference occurs. The Logic Tree is not displayed for Tags that do not allow referencing of other Tags (OPC, Virtual, etc...). The Logic Tree's visibility can be toggled by using the **View | Show Tag Detail** menu option.

## Configuration Entities

This section describes the configuration entities TrakSYS provides for modeling your production process.

### EVENTTrak Nodes

EVENTTrak nodes on the factory floor alert operators to recent and ongoing production events, and allow them to acknowledge, categorize, and add notes to important events. Each EVENTTrak in MODELTrak represents a running instance of the EVENTTrak control on a client node. In order for an EVENTTrak to receive event notifications, you must first assign it to the appropriate Event Definitions or Systems.

The EVENTTrak ActiveX controls must reside within a compatible ActiveX container, such as a Microsoft Visual Basic, C# or other HMI application. When you install the EVENTTrak control, the TrakSYS Setup automatically installs sample container applications which you may use as starting points for your customized EVENTTrak

container. Refer to the EVENTTrak section for more information on integrating the ActiveX controls into a client application.

### Properties

<b>General</b>	
Host Type	Specifies whether to identify the EVENTTrak node by Computer Name or IP Address.
Computer Name / IP Address	The Computer Name or IP Address of the EVENTTrak node.

## LOGICTrak Instances

In MODELTrak, a LOGICTrak instance represents a single instance of the LOGICTrak Service, which collects I/O, records KPI data, tracks production events, and dispatches event notifications to EVENTTrak controls. Refer to the LOGICTrak section for more information on both LOGICTrak Service and LOGICTrak Client.

### Properties

<b>General</b>	
Computer Name	Name of the computer where the LOGICTrak Service will run.
Scan (Milliseconds)	Target scan rate of the LOGICTrak in milliseconds, which determines how frequently LOGICTrak evaluates Tag changes and writes data to the database.
<b>Advanced</b>	
Script Class Name	Specifies the class name that will contain code that will run for this instance of LOGICTrak. The class name specified here must be implemented as an Advanced Script Library in the Administration section of MODELTrak.
Session Logging	Logs start and stop date/time information for the LOGICTrak Service.
Simulation Mode	Treats I/O Tags as Virtual Tags for troubleshooting purposes. LOGICTrak will not attempt to establish any connections to I/O servers.
Database Connect Attempts	The number of times LOGICTrak will attempt to connect to the database before abandoning the current scan. This setting may help alleviate transient connectivity problems. Note that this setting works in conjunction with the Database Manager Connect Timeout setting.
Virtual Tag Changed Seconds	When greater than 0, this setting instructs LOGICTrak to only examine Virtual Tags which have changed within the specified time limit. This optimizes LOGICTrak to disregard Virtual Tags which have not changed recently. Setting this value to 0 instructs LOGICTrak to examine all loaded Virtual Tags for possible value changes every scan.
Load All Tags	This setting instructs LOGICTrak to load every Tag in the database, regardless of whether or not the current configuration references all of them. LOGICTrak will still load the configuration and collect Event and KPI data normally.
Load Tags Only	This setting is similar to Load All Tags, but instructs LOGICTrak to ignore the configuration. When this setting is active, LOGICTrak will load every Tag in the database but will not load

	the configuration. Consequently, any State Tags dependent on configuration entities such as Event Definitions, Systems, and KPI Calculations will cease to function. A common use of this setting is to test I/O Tags before the configuration is complete, when the intent is not to collect any Event or KPI data.
<b>Tags</b>	
Tag	List of Tags LOGICTrak should load and monitor even if they are not specifically referenced by the configuration. By default, LOGICTrak only loads Tags referenced by the active configuration.

## SCRIBETrak Nodes

A SCRIBETrak node represents the installation of a SCRIBETrak forms application on a specific PC.

### Properties

<b>General</b>	
Host Type	Specifies whether to identify the SCRIBETrak node by Computer Name or IP Address.
Computer Name / IP Address	The Computer Name or IP Address of the SCRIBETrak node.
SCRIBETrak Item Groups	Contains the SCRIBETrak Item Groups that will be displayed on the specified node when the SCRIBETrak application is launched. SCRIBETrak Item Groups are defined in the SCRIBETrak section of MODELTrak.

## SCRIBETrak Item Group Assignments

Each SCRIBETrak Item Group Assignment links an Item Group to a SCRIBETrak node on a specific PC.

### Properties

<b>General</b>	
SCRIBETrak Item Group	Specifies which SCRIBETrak Item Group will be assigned to the SCRIBETrak node.
Show Header	Determines if the SCRIBETrak Item Group Name is displayed within the menu in the SCRIBETrak application on the specified node.

## Access Names

Access Names define how Tags connect to I/O servers or other data sources. TrakSYS provides a variety of built-in Access Name types for use with common data sources like OPC servers and OLEDB databases. Note that some Tags, such as Virtual Tags and State Tags, do not require Access Names. Access Names should be created prior to importing or creating Tags so that the Tags can be related to their appropriate Access Names they are created.

### Properties (OPC Access Name)

<b>General</b>	
Re-Connect (Seconds)	How often LOGICTrak should attempt to reconnect to the OPC Server if communication fails. Low values (i.e. 1 or 2 seconds) may make LOGICTrak unresponsive when lasting communication failures occur; the recommended setting is 60 seconds.
Quality Mask	Integer mask representing "good" quality. LOGICTrak verifies good qualities as follows: Tag Quality AND Quality Mask = Quality Mask. The OPC default is 192.
Simulation Mode	When checked, LOGICTrak will not attempt to connect to the actual data source, but instead will treat all of this Access Name's Tags as Virtual Tags.
Node	Name of the computer hosting the OPC server.
OPC Server	Specifies the the ProgID of the target OPC server (refer to your OPC server's documentation for more information). The browse button fills the list with the OPC server ProgIDs that are available on the server specified by the Node property above.
Access Path	A string prefix LOGICTrak will append to the front of every OPC item name. For example, if a complete OPC Item Name is [ML1500]N10:1, you can set this value to [ML1500] and then simply specify N10:1 as the OPC Tag's Item Name.
Read on Connect	When this property is checked, the LOGICTrak forces a read from all Tags under the Access Name when the connection is first made. OPC Servers typically return current values on connection even if this setting is not checked. This setting makes sure this happens even when the OPC server is not behaving as it should.
<b>Advanced</b>	
Use Alternate OPC Server	When checked, enables the use of an alternate fallback OPC server to use when communication with the primary OPC server fails. Ideally, such a backup OPC server should reside on a different computer than the primary OPC server.
Alternate Node	Name of the computer hosting the alternate OPC server.
Alternate OPC Server	ProgID of the alternate OPC server (refer to your OPC server's documentation for more information).
Alternate Access Path	String prefix for OPC item names when using the alternate OPC server.
Re-Connect While True	Used to manually trigger the OPC connection to be re-attempted, this setting specifies a Boolean condition by pointing at a discrete Tag. When True, the OPC connection will be closed and reopened.
Re-Connect Wait (Seconds)	Once a manual reconnect is triggered (see above), the LOGICTrak engine will wait this number of seconds before checking the trigger condition again.

### Properties (OPC HDA Access Name)

<b>General</b>	
Re-Connect (Seconds)	How often LOGICTrak should attempt to reconnect to the OPC Server if communication fails. Low values (i.e. 1 or 2 seconds) may make LOGICTrak unresponsive when lasting

	communication failures occur; the recommended setting is 60 seconds.
Quality Mask	Integer mask representing "good" quality. LOGICTrak verifies good qualities as follows: Tag Quality AND Quality Mask = Quality Mask. The OPC default is 192.
Simulation Mode	When checked, LOGICTrak will not attempt to connect to the actual data source, but instead will treat all of this Access Name's Tags as Virtual Tags.
Node	Name of the computer hosting the OPC server.
OPC Server	Specifies the the ProgID of the target OPC HDA server (refer to your OPC HDA server's documentation for more information). The browse button fills the list with the OPC HDA server ProgIDs that are available on the server specified by the Node property above.
Access Path	A string prefix LOGICTrak will append to the front of every OPC item name. For example, if a complete OPC Item Name is [ML1500]N10:1, you can set this value to [ML1500] and then simply specify N10:1 as the OPC Tag's Item Name.
Poll Interval (Seconds)	The delay LOGICTrak will wait before polling the data source. This interval determines how frequently LOGICTrak updates Tag values for this Access Name.
<b>Advanced</b>	
Re-Connect While True	Used to manually trigger the OPC HDA connection to be re-attempted, this setting specifies a Boolean condition by pointing at a discrete Tag. When True, the connection will be reattempted.
Re-Connect Wait (Seconds)	Once a manual reconnect is triggered (see above), the LOGICTrak engine will wait this number of seconds before checking the trigger condition again.

### Properties (Database Access Name)

<b>General</b>	
Re-Connect (Seconds)	How often LOGICTrak should attempt to reconnect to the database server if communication fails. Low values (i.e. 1 or 2 seconds) may make LOGICTrak unresponsive when lasting communication failures occur; the recommended setting is 60 seconds.
Quality Mask	Integer mask representing "good" quality. LOGICTrak verifies good qualities as follows: Tag Quality AND Quality Mask = Quality Mask. The default is 192.
Simulation Mode	When checked, LOGICTrak will not attempt to connect to the actual data source, but instead will treat all of this Access Name's Tags as Virtual Tags.
Connection String	OleDb connection string specifying how to connect to the database. This string should include the provider, initial catalog, data source, and security information.
Query	SQL query string for retrieving the desired data.
Query Interval (Seconds)	The delay LOGICTrak will wait before re-executing the specified Query. This interval determines how frequently LOGICTrak updates Database Tag values for this Access Name.

Query Timeout (Seconds)	Limit on how long LOGICTrak will wait for the specified Query to complete. If the Query times out, all dependent Database Tags will have bad Quality.
-------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------

### Database Access Name Queries

Database Access Names contain a SQL query to define the data points that are returned from the OLEDB connection. The query must conform to the rules and syntax of the query language used by the data source.

### Database Access Name Parameters

A database Access Name Parameter specifies a Tag value that can be passed into a query for use in the SQL statement.

#### Properties

General	
Name	The unique name given to the parameter. MODELTrak will automatically add the "@" character to the front of the Parameter Name after you enter it. For example, if the parameter in a SQL Server query is named "@Param", the Name property for the parameter can be entered as "Param" without the "@".
Tag	The TrakSYS Tag containing the value to be passed in this parameter. This Tag may be of any type (OPC Tag, Virtual Tag, Compare Tag, etc.).
Test Value	A default value to use when executing the database query by clicking on the "Test Query" button. This is used in place of the current value for the Tag for testing and debugging purposes.

### Tags

TrakSYS has four major categories of Tags: Logic Tags, I/O Tags, Virtual Tags, and State Tags. The sections below describe each of these Tag Types in more detail. The Properties below are common to all Tag types.

#### Properties

General	
Tag Group	Parent Tag Group for this Tag (solely for organizational purposes).
Data Type	The Tag's data type. Select String, Integer, Float, or Discrete. Some Tags do not support all of the above data types.
Default Value	Initial value for this Tag prior to its first evaluation. This value must be valid for the specified Data Type above. If the Tag Value Retention (below) is set then that value will supersede the Default Value when the LOGICTrak Service starts.
Tag Value Retention	Specifies that the Tag's value is saved to the database when the LOGICTrak Service stops and then reloaded as the default value when the LOGICTrak Service is restarted. The retained value will supersede the Default Value above when the LOGICTrak Service starts.
Persist Value Changes to	When checked, LOGICTrak will update the value of this Tag in the database whenever it changes.

Database	
<b>References</b>	
References	Lists all of the entities in TrakSYS that reference this Tag. A Tag cannot be deleted if it is referenced by other entities in the configuration.
<b>ALERTTrak</b>	
Enable Notification	Specifies that the Tag should report its value changes to ALERTTrak for subscriptions. This property is automatically set when a Tag is referenced by an ALERTTrak subscription.
<b>HISTORITrak</b>	
Historian	Lists all of the Tag History Definitions that have been created for this Tag. Tag History Definitions may be added, deleted, or edited from this section.
Lower Limit	Informational setting used when graphing historical trend data for a Tag in a report. This usually represents the lower control limit for a process variable or metric.
Upper Limit	Informational setting used when graphing historical trend data for a Tag in a report. This usually represents the upper control limit for a process variable or metric.
Lower Scale	Specifies the low value on a graph's Y-Axis scale when plotting historical data for a Tag. This setting ensures that all selected historical values for a Tag are visible within the boundaries of a chart for reporting purposes.
Upper Scale	Specifies the high value on a graph's Y-Axis scale when plotting historical data for a Tag. This setting ensures that all selected historical values for a Tag are visible within the boundaries of a chart for reporting purposes.

## Boolean Tags

Boolean Tags allow you to combine the values of two or more Tags using logical AND, OR, XOR, NAND, NOR, and XNOR operations. You can also specify what values to return if the Boolean operation evaluates to True or False, using either constants or Tags.

## Properties

<b>General</b>	
Operation	Boolean operation to perform on the specified list of Tags. Select AND, OR, XOR, NAND, NOR, or XNOR.
Tags	List of Tags for the Boolean operation. Boolean operations often only apply to two Tags, but LOGICTrak supports any number of Tags. Note that every Tag in this list must have a Discrete data type.
<b>Assign</b>	
Assign Tag	The target Tag whose value will be set based on the other Assign Tag properties. The Assign Tag is set when the Boolean Tag evaluates to True (see Assign Condition).
Assign Operation	When the Assign Tag is set, the Assign Value is applied to the Assign Tag using this operation.  <div style="display: flex; justify-content: space-between; padding: 0 20px;"> <span>=</span> <span>Assign Tag = Assign Value</span> </div> <div style="display: flex; justify-content: space-between; padding: 0 20px;"> <span>+</span> <span>Assign Tag = Assign Tag + Assign Value</span> </div>

	-           Assign Tag = Assign Tag - Assign Value *           Assign Tag = Assign Tag * Assign Value /           Assign Tag = Assign Tag / Assign Value %           Assign Tag = Assign Tag MOD Assign Value CONCAT   Assign Tag = Assign Tag & Assign Value
Assign Value	The value which will be set to the specified Assign Tag in conjunction with the Assign Operation.
Assign Condition	Determines if the Assign Tag assignment should occur once when the Boolean Tag first becomes True (On True) or constantly each scan while the Boolean Tag remains True (While True).
<b>Advanced</b>	
True Delay (Seconds)	The delay, in seconds, before the Tag's value becomes True once its expression evaluates to True.
False Delay (Seconds)	The delay, in seconds, before the Tag's value becomes False once its expression evaluates to False.
Tag Value if Compare is True	Value for the Tag to assume if its Boolean operation evaluates to True. This may be a constant or the value of another Tag.
Tag Value if Compare is False	Value for the Tag to assume if its Boolean operation evaluates to False. This may be a constant or the value of another Tag.
Prerequisite Tag	When set, LOGICTrak will always make sure that the referenced Tag is evaluated before this Tag. This is typically handled automatically based on references but there are some cases where this is necessary.
Force Evaluation	When checked, forces LOGICTrak to evaluate this Tag every scan even if no underlying Tags change. This setting is especially useful for Tags with circular references; without this setting, such Tags might never change.

## Calculation Tags

Calculation Tags perform simple arithmetic and string operations on two input Tags. Supported arithmetic operations include addition, subtraction, multiplication, division, modulus, and random number generation. String operations include concatenation and left and right substring extraction.

### Properties

<b>General</b>	
Left Operand	Value of the left (first) operand of the calculation, which may be a constant or the value of another Tag. This setting must have a valid Data Type for the specified Operation.
Operation	Calculation Operation to perform on the Left and Right Operands. Select + (addition), - (subtraction), * (multiplication), / (division), % (modulus), CONCAT (string concatenation), LEFT (left substring extraction), RIGHT (right substring extraction), or RANDOM (random number generation).
Right Operand	Value of the right (second) operand of the calculation, which may be a constant or the value of another Tag. This setting must have a valid Data Type for the specified Operation.
<b>Advanced</b>	
Prerequisite	When set, LOGICTrak will always make sure that the



	referenced Tag is evaluated before this Tag. This is typically handled automatically based on references but there are some cases where this is necessary.
Force Evaluation	When checked, forces LOGICTrak to evaluate this Tag every scan even if no underlying Tags change. This setting is especially useful for Tags with circular references; without this setting, such Tags might never change.

### Operations

Operation	Data Type	Description
+ (Add)	Integer or Float	Adds the value of the Left Operand and the Right Operand, and returns the result.
- (Subtract)	Integer or Float	Subtracts the value of the Right Operand from the value of the Left Operand, and returns the result.
* (Multiply)	Integer or Float	Multiplies the value of the Left Operand by the value of the Right Operand, and returns the result.
/ (Divide)	Integer or Float	Divides the value of the Left Operand by the value of the Right Operand, and returns the result.
% (Modulus)	Integer or Float	Divides the value of the Left Operand by the value of the Right Operand, and returns the remainder. Example: 10 % 3 = 1.
Concatenate	String	Concatenates (combines) the string value of the Left Operand with the string value of the Right Operand, and returns the result. Example: "thermo" CONCAT "former" = "thermoformer".
Left	Left Operand: String Right Operand: Integer Result: String	Extracts a substring from the string value of the Left Operand, using the leftmost number of characters specified by the value of the Right Operand. Example: LEFT("thermoformer", 6) = "thermo".
Right	Left Operand: String Right Operand: Integer Result: String	Extracts a substring from the string value of the Left Operand, using the rightmost number of characters specified by the value of the Right Operand. Example: RIGHT("thermoformer", 6) = "former".
Random	Integer or Float	Generates a random number every LOGICTrak Scan, using the value of the Left Operand as the minimum possible value and the value of the Right Operand as the maximum possible value.

## Compare Tags

Compare Tags compare two Tags and calculate a Discrete result. Available comparison operations include equality, inequality, greater than, greater than or equal to, less than, and less than or equal to. You can define what values to return when the comparison returns True or False, using either constants or Tags.

### Properties

<b>General</b>	
Left Operand	Value of the left (first) operand for the comparison, which may be a constant or the value of another Tag. This setting's data type must match the Right Operand's data type.
Operation	Comparison operation to perform on the Left and Right Operands. Select = (equal to), <> (not equal to), > (greater than), >= (greater than or equal to), < (less than), or <= (less than or equal to).
Right Operand	Value of the right (second) operand for the comparison, which may be a constant or the value of another Tag. This setting's data type must match the Left Operand's data type.
<b>Assign</b>	
Assign Tag	The target Tag whose value will be set based on the other Assign Tag properties. The Assign Tag is set when the Compare Tag evaluates to True (see Assign Condition).
Assign Operation	When the Assign Tag is set, the Assign Value is applied to the Assign Tag using this operation.  =            Assign Tag = Assign Value +            Assign Tag = Assign Tag + Assign Value -            Assign Tag = Assign Tag - Assign Value *            Assign Tag = Assign Tag * Assign Value /            Assign Tag = Assign Tag / Assign Value %            Assign Tag = Assign Tag MOD Assign Value CONCAT     Assign Tag = Assign Tag & Assign Value
Assign Value	The value which will be set to the specified Assign Tag in conjunction with the Assign Operation.
Assign Condition	Determines if the Assign Tag assignment should occur once when the Boolean Tag first becomes True (On True) or constantly each scan while the Boolean Tag remains True (While True).
<b>Advanced</b>	
True Delay (Seconds)	The delay, in seconds, before the Tag's value becomes True once its expression evaluates to True.
False Delay (Seconds)	The delay, in seconds, before the Tag's value becomes False once its expression evaluates to False.
Tag Value if Compare is True	Value for the Tag to assume if its Comparison operation evaluates to True. This may be a constant or the value of another Tag.
Tag Value if Compare is False	Value for the Tag to assume if its Comparison operation evaluates to False. This may be a constant or the value of another Tag.
Force Evaluation	When checked, forces LOGICTrak to evaluate this Tag every scan even if no underlying Tags change. This setting is especially useful for Tags with circular references; without this

	setting, such Tags might never change.
Prerequisite Tag	When set, LOGICTrak will always make sure that the referenced Tag is evaluated before this Tag. This is typically handled automatically based on references but there are some cases where this is necessary.
Force Evaluation	When checked, forces LOGICTrak to evaluate this Tag every scan even if no underlying Tags change. This setting is especially useful for Tags with circular references; without this setting, such Tags might never change.

## Counter Tags

Counter Tags allow a moving value to be monitored constantly by LOGICTrak and its changes to be accumulated within TrakSYS. The Counter Tag behaves similarly to the production counters that exist within KPI Calculation entities but can be used to count any incrementing value for the inclusion in logic or capture.

### Properties

<b>General</b>	
Counter	A Tag providing the Counter value.
Source Units Multiplier	Floating-point multiplier to convert Source Units to another unit of measurement.
Maximum Increment per Scan	The maximum number of (Counter) units the Counter may increment within a single scan. Set this to the maximum value the Counter can realistically increment to filter out unexpected large jumps in Counter values.
<b>Flow</b>	
Flow On Counter Units Count	Number of Counter Units that must pass through the Counter within the time specified by Flow On Period before the Counter is Flowing.
Flow On Period (Seconds)	A period of seconds during which the number of units specified by Flow On Counter Units Count must pass through the Counter before it is Flowing. Note that the Counter can begin Flowing sooner if it reaches Flow On Counter Units Count in less time than the Flow On Period.
Flow Off Delay (Seconds)	Delay after the Counter stops incrementing before it stops Flowing.
<b>Advanced</b>	
Reset On True	If this Tag is specified, the internal Counter value will reset to zero when the referenced Tag is True. If the Reset on True Tag remains zero, the Counter will continually reset effectively keeping the counter at zero.
Increment While True	If this Tag is specified, the Counter only accumulates differences in the Counter Tag movement while the referenced Tag is True.

## Latch Tags

Latch Tags are similar to Compare Tags, except that they use different conditions for setting and resetting. A Latch Tag has two states: "Set" and "Reset". A Latch Tag is

set when its Set Condition Tag evaluates to True, and is reset when its Reset Condition Tag evaluates to True.

An important distinction is that a Latch Tag will not automatically reset when its Set Condition Tag reverts to False, and will not automatically set when its Reset Condition Tag reverts to False. Once a Latch Tag sets, it remains set until its Reset Condition Tag evaluates to True; once a Latch Tag resets, it remains reset until its Set Condition Tag evaluates to True.

You can specify what values a Latch Tag should return for its Set and Reset states, using either Tags or constant values.

### Properties

<b>General</b>	
Latch Set Condition	A Boolean Tag whose value determines when the Latch should "Set." When True, the Latch will Set if it is currently Reset. When False, this Tag has no effect.
Set Value	The value for the Latch to return when it is in the Set state, which may be a constant or the value of another Tag. This setting's data type must match the Latch's Data Type setting.
Latch Reset Condition	A Boolean Tag whose value determines when the Latch should "Reset." When True, the Latch will Reset if it is currently Set. When False, this Tag has no effect.
Reset Value	The value for the Latch to return when it is in the Reset state, which may be a constant or the value of another Tag. This setting's data type must match the Latch's Data Type setting.
<b>Advanced</b>	
Force Evaluation	When checked, forces LOGICTrak to evaluate this Tag every scan even if no underlying Tags change. This setting is especially useful for Tags with circular references; without this setting, such Tags might never change.

### Database Tags

Database Tags obtain their value from an OLE DB source, such as a Microsoft SQL Server database. Database Tags use their Access Names to query the OLE DB data source for data, and then return a single value from the result set.

A Database Tag obtains its value from a single Value Column in the result set. For result sets with multiple rows, you can specify a Key Column and Key Value to isolate a specific row using any unique, identifying key.

### Properties

<b>General</b>	
Access Name	The Access Name defining this Tag's OLE DB data source, including the query necessary to retrieve its value.
Value Column Name	The name of the Column in the result set returned by the Database Access Name to use for this Tag's value.
Use Key Column	For result sets with multiple rows, set this to True to specify a Key Column Name which uniquely identifies each row and then specify a Key Column Value to select a row to use for this

	Tag's value.
Key Column Name	Name of a Column which uniquely identifies each row. Use this setting in combination with Key Column Value to isolate a single row from a result set with multiple rows. For example, if the OLE DB query returns a result set with an ID column uniquely identifying each row, set this property to "ID" and then set Key Column Value to the ID of the row you would like to use for this Tag's value. Use Key Column must be True for this setting to take effect.
Key Column Data Type	The data type of the Column specified by Key Column Name. For example, if your Key Column Name specifies an integer ID column, this setting should be Integer. Value options are String, Integer, Float, and Discrete.
Key Column Value	<p>The key value to isolate a single row in a result set which may contain multiple rows. Use Key Column Name to specify which column represents the result set's keys. LOGICTrak will scan the OLE DB result set for a row with this value in the column specified by Key Column Name, and then return the value in the column specified by Value Column Name.</p> <p>For example, if your result set has an ID column uniquely identifying each row, set Key Column Name to "ID" and set Key Column Value to the appropriate ID, such as 5, to narrow the results to a single row; LOGICTrak will then set the value of this Tag according to the column specified by Value Column Name.</p>

## OPC Tags

OPC Tags retrieve data from OPC-compliant I/O servers. The Item Name property specifies a unique, fully qualified address on the target OPC server. MODELTrak provides built-in support for Tag browsing, but not all OPC servers support browsing. Refer to your OPC server's documentation for more information.

TrakSYS also provides a wide range of options for fine-tuning and converting OPC values. You can mask out desired bits of an OPC value using any 32-bit mask and the following bitwise operations: AND, OR, and XOR. Deadband allows you to filter out small, unwanted fluctuations in Tag values. You can also convert an OPC value to an Engineering Units (EU) scale by specifying the Tag's Raw value range and the target EU value range.

Alternates provide a fallback mechanism in the event of I/O communication failure. When LOGICTrak detects that an OPC server has become inaccessible, it will automatically switch to the Access Name's alternate connection information, which may specify a backup OPC server on another node. Once it establishes a connection to the alternate OPC server, LOGICTrak will use the Alternate Item Names for the affected Tags.

## Properties

<b>General</b>	
Access Name	The Access Name specifying this Tag's OPC server.
Item Name	Unique, fully qualified address of an item (tag) on the OPC

	server.
Alternate Item Name	Alternate Item Name to use when the Access Name fails over to the Alternate OPC server. Refer to OPC Access Names for more information. If using alternates, this setting must be a unique, fully qualified address of an item (tag) on the alternate OPC server.
<b>Advanced</b>	
Mask Operation	The Operation LOGICTrak should use in conjunction with the Mask Value to discard (mask out) the undesired bits of an OPC item's value. Valid settings are None (no mask), AND, OR, and XOR. The Data Type setting for the result may be Integer, Float, or Discrete. If the target OPC Item is floating-point, LOGICTrak will truncate the value to an Integer before applying the mask. You cannot perform Scaling and Deadband operations in conjunction with a Mask operation.
Mask Value	<p>Integer Mask value to extract (mask out) the desired bits of an OPC item's value. LOGICTrak will combine this mask with the original OPC item's value to set the Tag value, using the operation specified by Mask Operation.</p> <p>For example, to isolate the 4<sup>th</sup> bit of an OPC item with a Mask, set the Mask Operation to AND and then set the Mask Value to 8. To isolate the first 4 bits of an OPC item, set the Mask Operation to AND and then set the Mask Value to 15 (1111 in binary).</p>
Use Scale	<p>When True, LOGICTrak will convert the raw OPC item value to a custom Engineering Units (EU) scale. Once you specify a Raw value range and an EU range, LOGICTrak can map the raw OPC values to their EU equivalents. Scaling is only available for Integer and Float Data Types, and always occurs before Deadband filtering. You cannot use Scaling in conjunction with a Mask.</p> <p>For example, if an OPC item's value always falls between 0 and 1000, and you would like to convert the value into a percent, specify 0 as the Scale Minimum Raw, 1000 as the Scale Maximum Raw, 0 as the Scale Minimum EU, and 100 as the Scale Maximum EU.</p>
Scale Minimum Raw	The minimum possible Raw value of the OPC item. Use Scale must be True to use this setting.
Scale Minimum EU	The minimum legal Engineering Units (EU) value. Use Scale must be True to use this setting.
Scale Maximum Raw	The maximum possible Raw value of the OPC item. Use Scale must be True to use this setting.
Scale Maximum EU	The maximum legal Engineering Units (EU) value. Use Scale must be True to use this setting.
Deadband	A floating-point setting for filtering out small, unwanted changes in the Tag value. After applying any applicable Scaling, LOGICTrak will ignore any Tag changes smaller or equal to this value. This setting is only applicable to Integer and Float Data Types, and is unavailable if you are using a Mask.

## OPC HDA Tags

OPC HDA Tags are references to data points that are configured in historian systems external to TrakSYS. By referencing HDA Tags in TrakSYS, their values are available for trending in select WEBTrak web parts without causing TrakSYS to re-record all of the actual Tag changes. OPC HDA Tags can be retrieved from external historians that are enabled with an OPC HDA server application.

### Properties

<b>General</b>	
Access Name	The Access Name specifying this Tag's OPC HDA server.
Item Name	Unique, fully qualified address of an item (tag) on the OPC HDA server.
<b>Advanced</b>	
Mask Operation	The Operation LOGICTrak should use in conjunction with the Mask Value to discard (mask out) the undesired bits of an OPC HDA item's value. Valid settings are None (no mask), AND, OR, and XOR. The Data Type setting for the result may be Integer, Float, or Discrete. If the target OPC HDA Item is floating-point, LOGICTrak will truncate the value to an Integer before applying the mask. You cannot perform Scaling and Deadband operations in conjunction with a Mask operation.
Mask Value	<p>Integer Mask value to extract (mask out) the desired bits of an OPC HDA item's value. LOGICTrak will combine this mask with the original OPC item's value to set the Tag value, using the operation specified by Mask Operation.</p> <p>For example, to isolate the 4<sup>th</sup> bit of an OPC HDA item with a Mask, set the Mask Operation to AND and then set the Mask Value to 8. To isolate the first 4 bits of an OPC item, set the Mask Operation to AND and then set the Mask Value to 15 (1111 in binary).</p>
Use Scale	<p>When True, LOGICTrak will convert the raw OPC HDA item value to a custom Engineering Units (EU) scale. Once you specify a Raw value range and an EU range, LOGICTrak can map the raw OPC HDA values to their EU equivalents. Scaling is only available for Integer and Float Data Types, and always occurs before Deadband filtering. You cannot use Scaling in conjunction with a Mask.</p> <p>For example, if an OPC HDA item's value always falls between 0 and 1000, and you would like to convert the value into a percent, specify 0 as the Scale Minimum Raw, 1000 as the Scale Maximum Raw, 0 as the Scale Minimum EU, and 100 as the Scale Maximum EU.</p>
Scale Minimum Raw	The minimum possible Raw value of the OPC HDA item. Use Scale must be True to use this setting.
Scale Minimum EU	The minimum legal Engineering Units (EU) value. Use Scale must be True to use this setting.
Scale Maximum Raw	The maximum possible Raw value of the OPC HDA item. Use Scale must be True to use this setting.

Scale Maximum EU	The maximum legal Engineering Units (EU) value. Use Scale must be True to use this setting.
Deadband	A floating-point setting for filtering out small, unwanted changes in the Tag value. After applying any applicable Scaling, LOGICTrak will ignore any Tag changes smaller or equal to this value. This setting is only applicable to Integer and Float Data Types, and is unavailable if you are using a Mask.

## State Tags

State Tags provide a means to access internal LOGICTrak state information, such as the status of a System, what Event Definitions are active, and how long LOGICTrak has been running. MODELTrak provides drop-down selection of all available state information in the system.

### Properties

<b>General</b>	
Entity Type	The type of Configuration Entity to obtain state information from. Use the drop-down to select this value.
Entity	Which Entity to obtain state information from. Once you specify an Entity Type, Configuration provides a drop-down list of the available entities or an appropriate picker.
Attribute	The Attribute of the selected Entity to use for the Tag's value. Refer to the State Tag Attribute Reference section for details.

### State Tag Attribute Reference

This section describes each available State Tag Attribute for every Entity Type. Any Custom Properties defined will also show up as available choices where applicable.

<b>Access Name</b>		
ID	Integer	Unique ID of this entity in the database.
Name	String	Display name of this entity.
Connected	Discrete	True if LOGICTrak has an open connection to the data source, False otherwise.
<b>Event Definition</b>		
ID	Integer	Unique ID of this entity in the database.
Name	String	Display name of this entity.
IsEventActive	Discrete	True if this Event Definition is active, False otherwise.
IsEventActiveCondition	Discrete	True if this Event Definition's Discrete Trigger Tag value is True, False otherwise.
NotEventActive	Discrete	True if this Event Definition is not active, False if it is active.
NotEventActiveCondition	Discrete	True if this Event Definition's Discrete Trigger Tag value is False, False if its Trigger Tag value is True.
OeeType	Integer	None = 0, Performance Loss = 1, Availability Loss = 2, System Not Scheduled = 3
SubSystemID	Integer	The ID of the corresponding Sub-System



		or -1 if this Event Definition does not belong to a Sub-System.
SubSystemName	String	The name of the corresponding Sub-System, or an empty string if this Event Definition does not belong to a Sub-System.
Shift	Shift	The current Shift instance for this Event Definition, typically corresponding to one of the Schedule Pattern Items in your Schedule.
<b>Shift</b>		
ShiftID	Integer	Unique ID of the corresponding Shift in the database.
ShiftName	String	Display name of the corresponding Shift entity.
TeamID	Integer	Unique ID of the corresponding Team in the database.
TeamName	String	Display name of the corresponding Team entity.
Scheduled	Discrete	True if this Shift instance represents Scheduled time (for KPI purposes), False otherwise.
NotScheduled	Discrete	True if this Shift instance represents unscheduled time (for KPI purposes), False if it represents Scheduled time.
StartDateTime	DateTime	Date and Time this Shift instance started.
EndDateTime	DateTime	Date and Time this Shift instance will end.
Date	DateTime	Date (at midnight) this Shift instance belongs to.
Duration	TimeSpan	The length of this Shift instance
Elapsed	TimeSpan	The amount of time that has elapsed since the beginning of this Shift instance.
Remaining	TimeSpan	The amount of time remaining in this Shift instance.
<b>LOGICTrak</b>		
ID	Integer	Unique ID of this entity in the database.
Name	String	Display name of this entity.
DateTime	DateTime	Current Date and Time on the machine where the LOGICTrak Service is running.
StartDateTime	DateTime	The Date and Time the LOGICTrak Service started.
RunningDuration	TimeSpan	How long the LOGICTrak Service has been running.
<b>KPI Calculation</b>		
ID	Integer	Unique ID of this entity in the database.
Name	String	Display name of this entity.
Flowing	Discrete	True if any of the KPI Inputs (Total, Good, or Bad) are Flowing, False otherwise. Refer to the section on KPI Counter Flow for more information.
NotFlowing	Discrete	Opposite of Flowing.
Total	KPI Input	The Total KPI Input.

Good	KPI Input	The Good KPI Input.
Bad	KPI Input	The Bad KPI Input.
Interval	KPI Interval	The current KPI Interval.
<b>KPI Input</b>		
ID	Integer	Unique ID of this entity in the database.
Flowing	Discrete	True if this Input is Flowing, False otherwise. Refer to the section on KPI Counter Flow for more information.
NotFlowing	Discrete	Opposite of Flowing.
FlowingDuration	TimeSpan	How long the Input has been Flowing or 0 if Flow is not active.
Count	Integer	The total number of units, converted to Calculation Units, that have passed through the Input during the current KPI Interval. Refer to the section on KPI Units for more information on OEE unit conversions.
Multiplier	Double	The active multiplier to convert Input Units to Calculation Units.
<b>KPI Interval</b>		
StartDateTime	DateTime	The Date and Time this Interval started.
Date	DateTime	Date (at midnight) this Interval belongs to.
Duration	TimeSpan	The amount of time that has elapsed since the beginning of this Interval.
<b>KPI Counter</b>		
ID	Integer	Unique ID of this entity in the database.
Name	String	Display name of this entity.
Flowing	Discrete	True if this Counter is Flowing, False otherwise. Refer to the section on KPI Counter Flow for more information.
NotFlowing	Discrete	Opposite of Flowing.
FlowingDuration	TimeSpan	How long the Counter has been Flowing or 0 if Flow is not active.
Count	Integer	The number of units, in raw Counter Units, that have passed through the Counter during the current KPI Interval. Refer to the section on KPI Units for more information on OEE unit conversions.
Multiplier	Double	The active multiplier to convert Counter Units to Input Units.
<b>Sample Definition</b>		
ID	Integer	Unique ID of this entity in the database.
Name	String	Display name of this entity.
ProcessMean	Float	The Process Mean configured for the specified Sample Definition.
ProcessLCL	Float	The Process Lower Control Limit configured for the specified Sample Definition.
ProcessUCL	Float	The Process Upper Control Limit configured for the specified Sample Definition.
ProcessLSL	Float	The Process Lower Specification Limit configured for the specified Sample

		Definition.
ProcessUSL	Float	The Process Upper Specification Limit configured for the specified Sample Definition.
<b>Schedule</b>		
ID	Integer	Unique ID of this entity in the database.
Name	String	Display name of this entity.
ReloadDateTime	DateTime	Last Date and Time LOGICTrak reloaded this Schedule from the database to check for changes. LOGICTrak reloads Schedules according to their ReloadInterval setting.
ReloadDuration	TimeSpan	The amount of time that has elapsed since ReloadDateTime.
CurrentShift	Shift	The current Shift instance for this Schedule, typically corresponding to one of the Schedule Pattern Items in your Schedule.
<b>System</b>		
ID	Integer	Unique ID of this entity in the database.
Name	String	Display name of this entity.
IsEventActiveCondition	Discrete	True if any of the Event Definitions' Discrete Trigger Tag values are True, False otherwise.
NotEventActiveCondition	Discrete	True if all of the Event Definitions' Discrete Trigger Tag values are False, False otherwise.
IsEventActive	Discrete	True if this System has an active Event Definition, False otherwise.
NotEventActive	Discrete	True if this System has no active Event Definitions, False if one or more Event Definitions are active.
EventActiveDuration	TimeSpan	How long the current System Event has been active, or 0 if there is no current System Event.
EventActiveEventDefinition	Event Definition	The root cause Event Definition of the current System Event.
<b>CurrentProduct</b>		
Name	String	Display name of this entity.
Code	String	Display the current product's code.
Description	String	Display description of this entity.
Attribute01 – Attribute20	String	Display the corresponding attribute value for the selected product.
<b>Tag</b>		
ID	Integer	Unique ID of this entity in the database.
Name	String	Display name of this entity.
Quality	Integer	The raw Quality associated with this Tag. Good Quality is typically 192.
HasFirstGoodValue	Discrete	True if LOGICTrak has acquired the first good Quality value of this Tag from its data source, False otherwise.
LastValue	String	The last value of this Tag.
LastValueString	String	The last value of this Tag converted to a

		String Data Type.
LastValueInteger	Integer	The last value of this Tag converted to a Integer Data Type.
LastValueFloat	Float	The last value of this Tag converted to a Float Data Type.
LastValueDiscrete	Discrete	The last value of this Tag converted to a Discrete Data Type.
ValueChangedDuration	TimeSpan	The amount of time that has elapsed since this Tag's value last changed.
Flowing	Discrete	(Applies only to Counter Tags) True if this Counter is Flowing, False otherwise. Refer to the section on KPI Counter Flow for more information.
NotFlowing	Discrete	(Applies only to Counter Tags) Opposite of Flowing.
FlowDuration	TimeSpan	(Applies only to Counter Tags) How long the Counter has been Flowing or 0 if Flow is not active.
OpcWriteStatusValue	String	(Applies only to OPC Tags) The last value that was written to the Tag.
OpcWriteStatusDateTime	DateTime	(Applies only to OPC Tags) The last date/time that a value was written to the Tag.
<b>DateTime</b>		
String	String	String representation of this Date and Time.
Hour	Integer	Hour component of the Time.
Minute	Integer	Minute component of the Time.
Second	Integer	Second component of the Time.
Year	Integer	Year component of the Date.
Month	Integer	Month component of the Date.
Day	Integer	Day component of the Date (i.e. 1-31 on a calendar month).
DayOfWeek	Integer	This Date's day of the week. Sunday = 0, Monday = 1, Tuesday = 2, Wednesday = 3, Thursday = 4, Friday = 5, Saturday = 6.
<b>TimeSpan</b>		
Seconds	Integer	The whole number of seconds in this time span. For example, if the TimeSpan is 70 seconds long, this value will be 10 seconds.
Minutes	Integer	The whole number of minutes in this time span. For example, if the TimeSpan is 75 minutes long, this value will be 15 minutes.
Hours	Integer	The whole number of hours in this time span. For example, if the TimeSpan is 26 hours long, this value will be 2 hours.
Days	Integer	The whole number of days in this time span. For example, if the TimeSpan is 60 hours long, this value will be 2 days.
TotalSeconds	Double	The total number of seconds in this time

		span. For example, if the TimeSpan is one hour long, this value will be 3600 total seconds.
TotalMinutes	Double	The total number of minutes in this time span. For example, if the TimeSpan is 1.5 hours, this value will be 90 total minutes.
TotalHours	Double	The total number of hours in this time span. For example, if the TimeSpan is 1.5 days, this value will be 36 total hours.
TotalDays	Double	The total number of days in this time span. For example, if the TimeSpan is 36 hours, this value will be 1.5 total days.

## Switch Tags

Switch Tags define complex conditional logic using a single Switch Tag and a set of Cases. If you need to compare a Tag to many possible values, you can substitute a Switch Tag for many Compare Tags. Each Switch Tag defines a set of Cases, which LOGICTrak compares to the Switch Tag setting for equality; a Case specifies a constant or Tag to compare to the Switch Tag and a Value constant or Tag to return if the Case matches. You can specify a Switch Default to use as the Tag's return value if there is no matching Case; otherwise the Tag retains its previous value.

## Properties

<b>General</b>	
Switch Tag	Tag to use as the "Switch." Every scan, LOGICTrak compares the value of the Switch Tag to each Case Tag. If it finds a match, this Tag returns the value of the matching Case Value; if LOGICTrak does not find a match, it sets this Tag's value according to the Use Default setting.
Use Default	When checked, LOGICTrak will use the Switch Default setting to set the Tag value if no Cases match the Switch Tag's value. Otherwise, the Tag retains its previous value when there is no matching Case.
Switch Default	Constant or Tag value to use as the Tag's value when no Cases match the Switch Tag value. This setting is only available when Use Default is checked.
Cases	List of Case and Value pairs to compare to the Switch Tag.
Case (Case Properties)	Constant or Tag value to compare to the Switch Tag. If the case matches, LOGICTrak will use the corresponding Case Value to set the Tag.
Value (Case Properties)	Constant or Tag value to use as the Tag's value if its Case matches the Switch Tag. This value must be valid for the specified Data Type.
<b>Advanced</b>	
Force Evaluation	When checked, forces LOGICTrak to evaluate this Tag every scan even if no underlying Tags change. This setting is especially useful for Tags with circular references; without this setting, such Tags might never change.

## Virtual Tags

Virtual Tags function as internal system variables. You can use Virtual Tags to store any information that you find useful. A common use of Virtual Tags is to store active Product and KPI information. EVENTTrak provides an interface for modifying Virtual Tags while the system is running.

### Properties

General	
Value	The current value of this Virtual Tag.

## Script Tags

Script Tags allow simple and complex expressions and calculations to be created in a single Tag using .NET scripting/programming languages. There are two types of Script Tags, Simple and Advanced.

Simple Script Tag expressions are created using a VB.NET syntax. Any VB.NET language features may be used but the expression must be a single line (not physically) of code that evaluates to a single value of one of the 4 TrakSYS Tag data types (String, Integer, Float, Discrete).

Advanced Script Tag expressions are created using a C#.NET syntax. The Advanced Script Tag expression can be multiple lines of code but must end with a C# return statement which returns a single value of one of the 4 TrakSYS Tag data types (String, Integer, Float, and Discrete).

The Advanced type is only available if the Advanced Script Tag feature is licensed.

### Properties

General	
Script Type	Simple or Advanced (see descriptions above).
Script	Contains the expression script for this Tag.
Advanced	
Force Evaluation	When checked, forces LOGICTrak to evaluate this Tag every scan even if no underlying Tags change. This setting is especially useful for Tags with circular references; without this setting, such Tags might never change.
Auto Generate Prerequisites	When checked, any Tags that are referenced within the script are automatically added to this Tag's internal reference list. This value should only be unchecked when writing Advanced Script Tags that reference other Tag values by dynamically calculating their names.
Prerequisites	Lists all of the Tags that are referenced by this Script Tag. This list is automatically maintained unless the Auto Generate Prerequisites option (above) is unchecked.

## Aggregate Tags

Aggregate Tags allow for common operations to be made on a list of Tags.

### Properties

General	
Operation	Specifies the operation that is performed on the list of Tags

	defined in the Aggregate Tag. Options include Sum, Average, Min and Max.
<b>Advanced</b>	
Prerequisite Tag	When set, LOGICTrak will always make sure that the referenced Tag is evaluated before this Tag. This is typically handled automatically based on references but there are some cases where this is necessary.
Force Evaluation	When checked, forces LOGICTrak to evaluate this Tag every scan even if no underlying Tags change. This setting is especially useful for Tags with circular references; without this setting, such Tags might never change.

## Template Tags

The Template Tag is a special Tag type used as a placeholder for Tag properties in Template entities. For instance, when creating a Template Event Definition, the Trigger Tag can be set to a Template Tag instead of a real Tag. Besides being a non-functional Tag placeholder, the Template Tag also works in conjunction with the automatic Tag assignment functionality of Template to child entities. When using the Template Tag Prefix property in the Advanced tab of Template and child Systems, the Template Tag is available to be automatically replaced with a related real Tag when the automatic Tag assignment operation is run.

For example, A Template System is created called Template Bottle Line. Its Template Tag Prefix is configured as **TBL\_**. A Template Tag for the Product is created and named **TBL\_Product** and assigned to the System. A child System called Bottle Line 1 is created. Its Template Tag Prefix is configured as **BL1\_**. A Virtual Tag is created for the Bottle Line 1 Product and called **BL1\_Product** but not assigned to the System. When the Assign Tags from Template Parent operation is run on Bottle Line 1, the Template Tag Prefix settings are used and each Template Tag assigned to the parent triggers a search for a similarly named Tag (with the exception of the prefix). Tag matches are found and assigned to Bottle Line 1. The **BL1\_Product Tag** is located and automatically assigned as the Product Tag to Bottle Line 1.

## Tag Views

Tag Views allow all of the Tags in TrakSYS to be browsed by their Tag Type or Data Type.

## Tag History Definitions

A Tag History Definition defines the criteria and time intervals for recording Tag values by TrakSYS for historical analysis and reporting. Multiple History Definitions can be created for a Tag in order to support different visualizations and reporting requirements of the data.

### Properties

<b>General</b>	
Mode	Defines how LOGICTrak will record Tag values.  Periodic : Records Tag value changes on a fixed time interval.  On Change (Simple) : Records the Tag value when it changes

	<p>but only records the latest Tag value as of the current Scan time.</p> <p>Periodic and On Change (Simple) : A combination of both the Periodic and On Change (Simple) modes above.</p> <p>On Change (Detailed) : This mode applies to OPC Tags only. Records the Tag value on every change. When a Tag value changes multiple times between scans, this mode records every change.</p> <p>Manual : Does not automatically record Tag value changes.</p> <p>The Manual mode is used when an outside application (such as EVENTTrak) is being used to populate the Tag History.</p>
Interval (Seconds)	The time interval, in seconds, at which LOGICTrak will record the current value of the Tag. This setting is only applicable to History Definitions with a Mode value of Periodic or Periodic and On Change.
Expiration (Hours)	The amount of time, in hours, that TrakSYS will maintain a record of a Tag's historical value in the database prior to deleting it. This setting allows TrakSYS to automatically purge old data that is no longer needed for analysis. Tag History data will never expire when this is set to a value of zero (0).
Deadband	A floating-point setting for filtering out small, unwanted changes in the Tag value. LOGICTrak will require the change in a Tag value to be greater than this setting before recording the value to the Tag history. This setting is only applicable to History Definitions with a Mode value of On Change or Periodic and On Change.
LOGICTrak	The parent LOGICTrak which records historical data for this Tag. If there is only one configured LOGICTrak, this setting will default to that LOGICTrak.

## Areas

Each LOGICTrak monitors one or more Areas, which may represent any subset of your production environment. An Area may represent a plant, a manufacturing line, a single piece of equipment, or anything else. Areas contain Systems, but they may also contain other Areas, allowing you to hierarchically model your process.

## Properties

<b>General</b>	
Area Type	An Area Type categorizing this Area (solely for organizational and reporting purposes).
S95 Area Type	This specifies type of Area in terms of the S95 model. The available options are Enterprise, Site and Area.
LOGICTrak	The parent LOGICTrak which evaluates this Area. If there is only one configured LOGICTrak, this setting will default to that LOGICTrak.
Show in Reports	Determines if this entity is displayed in report pickers.



## Systems

Systems typically model part of a manufacturing line, such as a packaging or production machine. They are the basis for Event tracking, Product Management, Shift Schedules, EVENTTrak notification, and KPI collection.

Systems appear in Areas and Systems section in the MODELTrak tree view, and appear in the list view when you select their parent Area.

There are two types of Systems in TrakSYS: Discrete Systems (or Production Lines in S95) and Batch Systems (or Process Cells in S95). The sections below describe each of these System types in more detail. The Properties below are common to both System types. Please also see the section for System Templates.

### Properties

General	
External ID	The External ID is used to relate this entity to a matching item in an external System for purposes of custom interoperability. For instance, it could be the unique identifier for this entity in an ERP or other business application. This property is not used internally in TrakSYS for any standard functionality
System Type	A System Type categorizing this System (solely for organizational and reporting purposes).
Schedule	The Schedule this System uses for Shifts.
Split Events on Day Change	When checked, LOGICTrak will split any active Events for this System at midnight. This may help certain reports associate Events with the proper dates, but it will also artificially increase the reported number of Events and Sub-Events.
Split Events on Shift Change	When checked, LOGICTrak will split any active Events for this System when a Shift change occurs. This may be necessary to associate Events with the proper Shift for reporting, but it will also artificially increase the reported number of Events and Sub-Events.
Split Events on Product Change	When checked, LOGICTrak will split any active Events for this System when a Product change occurs. Without this enabled, Product changes will not take effect until the next Event begins and the product-based reports will not be accurate. Note that using this setting will artificially increase the reported number of Events and Sub-Events.
Split Events on job Change	When checked, LOGICTrak will split any active Events for this System when a Job change occurs. Without this enabled, Job changes will not take effect until the next Event begins and the product-based reports will not be accurate. Note that using this setting will artificially increase the reported number of Events and Sub-Events.
Product Set	The set of Products that run on this System.
Job Mode	Determines when Job records are created in the database when the Job Tag changes to a new value. The <b>Use Existing Jobs</b> option restricts TrakSYS to only look at the Job table and relate the System with a Job if it currently exists. The <b>Create New Jobs</b> option will first look for existing Job records but if a match is not found, a new Job record will be created.

Show in Reports	Determines if this entity is displayed in report pickers.
<b>Job Capture</b>	
Job Capture Scheme	Specifies the Job Capture Scheme to be used to snapshot Tag values when a new Job is created or assigned.
Job Capture Scheme Columns	Allows specific Tags to be assigned to the Job Capture Scheme columns for this System.
<b>EVENTTrak</b>	
EVENTTrak Nodes	List of EVENTTrak nodes that will receive Event notifications from the LOGICTrak for this System.
Synchronize	Copies the selected EVENTTrak's settings to every other EVENTTrak in the list.
<b>Categories</b>	
Available Event Category Groups	List of available Event Categories you may assign to this System.
Assigned Event Category Groups	List of Event Categories Groups that will appear in EVENTTrak for operator selection when an Event occurs on this System.
Event Category Scheme	Scheme defining each level of Event Categories assigned to this System. The reports will use the specified Event Category Scheme to correctly label Event Category levels (i.e. Reason, Sub-Reason, etc.).
<b>VTRs</b>	
VTRs	List of Tags to snapshot when an Event starts or ends. The values of these Tags will be available in the Event Detail view of Events Reports for this System.
<b>Tags</b>	
Tags	List of Tags related to this System. These relationships may be used in reporting to list or trend Tags based on the selected System.
<b>Advanced</b>	
Script Class Name	Specifies the class name that will contain code that will run for this System. The class name specified here must be implemented as an Advanced Script Library in the Administration section of MODELTrak.
Template Tag Prefix	Specifies the beginning characters for Tags that will be used and assigned to this System and its underlying configuration. See the Template Tag section for more details on how the Template Tag Prefix is used.
Event Date	What date LOGICTrak should assign to Event records as they are recorded. This setting is disabled in TrakSYS 7 and is set to "Event Shift Date" to use the date assigned to the current Shift in progress for the parent System.
Enable Notification	Specifies that the System should report its state changes to ALERTTrak for subscriptions. This property is automatically set when a System is referenced by an ALERTTrak subscription.

## Discrete Systems

Discrete Systems (Production Lines in S95) contain sets of Event Definitions specifying what Events to track. They may also contain Sub-Systems to organize the Event Definitions into groups based on smaller components within the main System.

## Properties

<b>General</b>	
Product Code Tag	Tag this System uses to select its current Product from the Product Set. The Product Code Tag value must uniquely identify whatever Product is running.
Job Tag	The Tag specified here holds a string that identifies the current Job that is running on the System.
Job Planned Size Tag	If a Tag is specified here, its value is captured and copied to the Planned Size field for a Job when the Job starts.
<b>System Capture</b>	
System Capture Scheme	System Capture Scheme defining the Capture Scheme Columns for this System.
System Capture Scheme Columns	List of Capture Tags for the selected Capture Scheme Columns. Double-click an entry to select a Capture Tag for the corresponding Column.
<b>SPCTrak</b>	
Sample Definitions	Lists the configured Sample Definitions for this System.
<b>Advanced</b>	
Impact Tag (0 to 1)	Impact determines how to weight Events for this System. LOGICTrak will record the value of the Impact Tag at the start of each Event as weight. Certain reports use these weights to display weighted durations of Events. For example, if an Event lasts 10 minutes, but has an Impact of 0.5, the weighted duration is 5 minutes.
Suppress Events If True	When set, the value of Discrete Tag is added to the logic of all child Event Definitions. While the Suppress Tag is True, none of the child Event Definitions will evaluate to True. The Event Definition properties expose a property for each Event Definition that may override this behavior.
Manual Data Entry	When checked, specifies that the System is a "Manual" System. Manual Systems are not loaded by LOGICTrak and their configuration is supports manual data entry via the SCRIBETrak application forms. Manual Systems must have exactly one KPI Calculation assigned to them. Certain System, Event Definition and KPI Calculation properties do not apply to Manual Systems and are hidden in the property dialogs.

## Batch Systems

Batch Systems (Process Cells in S95) contain sets of Function Definitions specifying the potential steps for the process in a batching environment. They contain Sub-Systems to organize the Function Definitions into groups based on smaller components within the parent System. Unlike a Discrete System, a Batch System must contain at least one Sub-System. In addition, Event and Function Definitions may not be defined under the parent System (they must be defined under the Sub-Systems).

## Properties

<b>General</b>	
Split Events on Batch Changes	When checked, LOGICTrak will split any active Events for this System when a Job change occurs. Without this enabled, Job changes will not take effect until the next Event begins and the job-based reports will not be accurate. Note that using this setting will artificially increase the reported number of Events and Sub-Events.
Batch Planned Size Tag	If a Tag is specified here, its value is captured and copied to the Planned Size field for a Batch when the Batch starts.
Batch Actual Size Tag	If a Tag is specified here, its value is captured and copied to the Actual Size field for a Batch when the Batch ends.
<b>Categories</b>	
Available Step Category Groups	List of available Step Categories you may assign to this System.
Assigned Step Category Groups	List of Step Categories Groups that will appear in EVENTTrak for operator selection when an Event occurs on this System.
Step Category Scheme	Scheme defining each level of Step Categories assigned to this System. The reports will use the specified Step Category Scheme to correctly label Step Category levels.

## System EVENTTrak Node Definitions

Represent EVENTTrak nodes assigned to a System, with additional settings for operator interaction. EVENTTrak nodes cannot be assigned at the System level if they are already assigned at the Sub-System or Event Definition level.

## Properties

<b>General</b>	
EVENTTrak	The EVENTTrak node assigned to this System.
Display Delay (Seconds)	Number of seconds after an Event begins before it appears in EVENTTrak.
Acknowledge Delay (Minutes)	Number of minutes operators have to acknowledge an Event before EVENTTrak auto-acknowledges it.
Allow Acknowledge While Active	When checked, allows operators to acknowledge and categorize Events while they are still active. This behavior requires custom programming with the EVENTTrak controls.
Force Acknowledge On Event End	When checked, EVENTTrak will automatically acknowledge Events as soon as they end.
Require Event Category	When checked, this requires operators to select an Event Category in order to acknowledge an Event. This behavior requires custom programming with the EVENTTrak controls.
Read Only	When checked, this System's Events will appear read-only in the EVENTTrak. Operators will not be able to

	acknowledge or categorize Events marked as Read Only. This behavior requires custom programming with the EVENTTrak controls.
<b>Message Data</b>	
Message Data	Field for attaching extra information to each Event notification LOGICTrak sends to the EVENTTrak (for this System).

## Job Capture Schemes

A Job Capture Scheme defines the Job Capture Scheme Columns associated with a Job. Once you assign Job Capture Tags to your Job's Capture Scheme Columns, LOGICTrak will record the values of the Capture Tags as supplementary data in matching columns for each Event. Capture Tags work similarly to VTRs, but the reports allow you to group and filter by Capture Columns rather than merely view the captured values in a detail view.

### Job Capture Scheme Columns

#### Properties

<b>General</b>	
Data Type	The Data Type of this Capture Column. Any Job Capture Tags associated with this Column must have the same Data Type.
Capture Value On	The time at which LOGICTrak should record the associated Capture Tags' values. Select "Job Start" for LOGICTrak to record the Capture Tags' values when a Job starts, or select "Job End" for LOGICTrak to record the Capture Tags' values when a job ends. If "Job Record Creation" is selected, data captures is done only on the initial Job start, but not on successive Job restarts. If N/A is selected no data capture is performed.

### Job Capture Tags

#### Properties

<b>General</b>	
Capture Source	The Tag for LOGICTrak to record for the corresponding Job Capture Scheme Column.
Name	The name of the corresponding Job Capture Scheme Column.
Data Type	The Data Type of the Job Capture Scheme Column. The Column and Capture Tag's Data Types must match.

## System Capture Schemes

A System Capture Scheme defines the System Capture Scheme Columns associated with a System. Once you assign System Capture Tags to your System's Capture Scheme Columns, LOGICTrak will record the values of the Capture Tags as

supplementary data in matching columns for each Event. Capture Tags work similarly to VTRs, but the reports allow you to group and filter by Capture Columns rather than merely view the captured values in a detail view. Examples of common Capture Columns are Job, Batch, and Lot Number.

## System Capture Scheme Columns

### Properties

<b>General</b>	
Data Type	The Data Type of this Capture Column. Any System Capture Tags associated with this Column must have the same Data Type.
Capture on Event Edge	The time at which LOGICTrak should record the associated Capture Tags' values. Select "Event Start" for LOGICTrak to record the Capture Tags' values when an Event starts, or select "Event End" for LOGICTrak to record the Capture Tags' values when an Event ends.
Split Event on Value Change	When checked, LOGICTrak will split the current Event whenever the associated Capture Tag's value changes. This setting is necessary if you wish to report by this Column; without it, there would be no record of when the Capture Tag changed.

## System Capture Tags

### Properties

<b>General</b>	
Capture Source	The Tag for LOGICTrak to record for the corresponding Capture Scheme Column.
Name	The name of the corresponding Capture Scheme Column.
Data Type	The Data Type of the corresponding Capture Scheme Column. The Column and Capture Tag's Data Types must match.
Event Edge Type	The Event Edge Type of the corresponding System Capture Scheme Column.
Split Event on Value Change	The Split Event on Value Change setting of the corresponding Capture Scheme Column.

## System VTR Definitions

System VTR (Value-To-Record) Definitions specify supplementary Tag Values to record when a System Event starts or ends.

### Properties

<b>General</b>	
Record Tag	Tag whose value LOGICTrak should record when an Event occurs.
Event Edge	The time at which LOGICTrak should record the VTR. Select "Event Start" for LOGICTrak to record the VTR when an Event starts, or select "Event End" for LOGICTrak to record the VTR when an Event ends.

## System Tag Relationships

System Tag Relationships provide a way to associate Tags with a System. There is no data collected or stored automatically as a result of configuring Tag Relationships, they exist for extensibility purposes only (for instance creating a report that trends a set of Tags based on the selected System).

### Properties

<b>General</b>	
Relate Tag	The Tag that will be associated with this entity.
Abbreviation	A shorter name or abbreviation for the related Tag.

## System Views

System Views are customizable lists of Systems which allow you to use any arbitrary criteria to relate and sort groups of Systems. Some reports offer System View filters, allowing you additional flexibility in choosing which Systems to include.

### Properties

<b>General</b>	
Available Systems	List of Systems available to add to this System View. Select a System from this list and then click Add to add it to the list of Assigned Systems.
Assigned Systems	List of Systems assigned to this System View. Systems in this list will appear in the view.

## Sample Definitions

A Sample Definition is a placeholder for SPC values to be recorded. A System may contain many Sample Definitions. Data for a Sample Definitions is added to the database using the API exposed in EVENTTrak or using the Advanced Script Tag API in the Script Tags or the Script Libraries.

### Properties

<b>General</b>	
Calculation Type	<p>Specifies how data is stored and calculated for the Sample Definitions.</p> <p>The Average mode supports storing some number of samples in a sub-group where the sample values are averaged to find the mean.</p> <p>The Percent mode assumes that the sub-group entries are bad counts for different reasons. A total number of sampled items</p>

	is a requirement for each sub-group so that a “percentage bad” can be calculated for each reason. The reasons are defined as Sample Categories.
Process Mean	Specifies the expected Mean for the process. This value may be used in plotting the SPC data.
Process Lower Control Limit (LCL)	Specifies the expected Lower Control Limit for the process. This value may be used in plotting the SPC data.
Process Upper Control Limit (UCL)	Specifies the expected Upper Control Limit for the process. This value may be used in plotting the SPC data.
Process Lower Specification Limit (LSL)	Specifies the expected Lower Specification Limit for the process. This value may be used in plotting the SPC data.
Process Upper Specification Limit (USL)	Specifies the expected Upper Specification Limit for the process. This value may be used in plotting the SPC data.

## Sub-Systems

Sub-Systems provide an additional level of grouping within normal Systems. This is useful for configuring Overview Systems; in these cases, Systems typically represent entire production lines while Sub-Systems represent equipment. With such a hierarchy you can easily relate line stoppages to the equipment and Event Definitions that caused them. The reports allow you to group your data by Sub-System, providing a clear view of the root cause for each interruption.

Sub-Systems appear beneath Systems in the MODELTrak tree view, and appear in the list view when you select their parent System.

There are two types of Sub-Systems in TrakSYS: Discrete Sub-Systems (or Work Cells in S95) and Batch Sub-Systems (or Units in S95). The sections below describe each of these Sub-System types in more detail. The Properties below are common to both Sub-System types.

## Properties

<b>General</b>	
System Type	A System Type categorizing this Sub-System (solely for organizational and reporting purposes).
Show in Reports	Determines if this entity is displayed in report pickers.
<b>EVENTTrak</b>	
EVENTTrak Nodes	List of EVENTTrak nodes that will receive Event notifications from the LOGICTrak for this Sub-System.
Synchronize	Copies the selected EVENTTrak's settings to every other EVENTTrak in the list.
<b>Categories</b>	
Available Event Category Groups	List of available Event Categories you may assign to this Sub-System.
Assigned Event Category Groups	List of Event Category Groups that will appear in EVENTTrak for operator selection when an Event occurs on this Sub-System.
<b>VTRs</b>	
VTRs	List of Tags to snapshot when an Event starts or ends. The values of these Tags will be available in a detail view of a report for this Sub-System.



<b>Tags</b>	
Tags	List of Tags related to this Sub-System. These relationships may be used in reporting to list or trend Tags based on the selected System.

## Discrete Sub-Systems

### Properties

<b>General</b>	
Linked Systems	When set, any event that is recorded under this Sub-System is also recorded with a link to the event that is currently active in the linked System. Reports allow the Event Definition from the linked System to be displayed in addition to the actual Event Definition.

## Batch Sub-Systems

### Properties

<b>General</b>	
Product Code Tag	The Tag this Sub-System uses to select its current Product from the Product Set. The Product Code Tag value must uniquely identify whatever Product is running.
Job Tag	The Tag specified here holds a string that identifies the current Job that is running on the Sub-System.
Batch Tag	The Tag specified here holds a string that identifies the current Batch that is running on the Sub-System.
<b>System Capture</b>	
System Capture Scheme	System Capture Scheme defining the Capture Scheme Columns for this Sub_System.
System Capture Scheme Columns	List of Capture Tags for the selected Capture Scheme Columns. Double-click an entry to select a Capture Tag for the corresponding Column.
<b>Categories</b>	
Available Step Category Groups	List of available Step Categories you may assign to this Sub-System.
Assigned Step Category Groups	List of Step Category Groups that will appear for operator selection when a Function Definition exceeds its target duration this Sub-System.
<b>SPCTrak</b>	
Sample Definitions	Lists the configured Sample Definitions for this Sub-System.
<b>Parameters</b>	
Parameters	List of Parameter Definitions associated with this Sub-System.
<b>Advanced</b>	
Script Class Name	Specifies the class name that will contain code that will run for this Sub-System. The class name specified here must be implemented as an Advanced Script Library in the Administration section of MODELTrak.
Impact Tag (0 to 1)	Impact determines how to weight Events for this Sub-System. LOGICTrak will record the value of the Impact Tag at the start of

	each Event as weight. Certain reports use these weights to display weighted durations of Events. For example, if an Event lasts 10 minutes, but has an Impact of 0.5, the weighted duration is 5 minutes.
Suppress Events If True	When set, the value of Discrete Tag is added to the logic of all child Event Definitions. While the Suppress Tag is True, none of the child Event Definitions will evaluate to True. The Event Definition properties expose a property for each Event Definition that may override this behavior.

## Batch Sub-System Parameter Definitions

Parameter Definitions allow the specification of a value that is to be recorded with the Batch or specific Batch Step. The actual value that is recorded is based on configuration set at the Recipe that is associated with the Batch.

### Properties

General	
Parameter Definition Type	A Parameter Definition Type categorizing this Parameter Definition (solely for organizational and reporting purposes).
Data Type	The Parameter's data type. Select String, Integer, Float, or Discrete.
Default Value	Initial value for this Parameter prior to its first evaluation. This value must be valid for Parameter's Data Type.
Parameter Tag	Initial value for this Parameter prior to its first evaluation. This value must be valid for Parameter's Data Type. If specified, the Parameter value from the Recipe is written to this Tag. This Tag must be an OPC or Virtual Tag. OPC Tags can only be written to if the license allows it.
Minimum Value	The minimum value that can be configured in the Recipe for this Parameter.
Maximum Value	The maximum value that can be configured in the Recipe for this Parameter.
Exception Value	The value that will be assigned to this Parameter when a Batch Step gets triggered in an unpredicted order.

## Sub-System EVENTTrak Node Definitions

Represent EVENTTrak nodes assigned to a Sub-System, with additional settings for operator interaction. EVENTTrak nodes cannot be assigned at the Sub-System level if they are already assigned at the System or Event Definition level.

### Properties

General	
EVENTTrak Nodes	The EVENTTrak nodes assigned to this Sub-System.

Display Delay	Number of seconds after an Event begins before it appears in EVENTTrak.
Acknowledge Delay	Number of minutes operators have to acknowledge an Event before EVENTTrak auto-acknowledges it.
Allow Acknowledge While Active	When checked, allows operators to acknowledge and categorize Events while they are still active. This behavior requires custom programming with the EVENTTrak controls.
Force Acknowledge On Event End	When checked, EVENTTrak will automatically acknowledge Events as soon as they end.
Require Event Category	When checked, this requires operators to select an Event Category in order to acknowledge an Event. This behavior requires custom programming with the EVENTTrak controls.
Read Only	When checked, this Sub-System's Events will appear read-only in the EVENTTrak. Operators will not be able to acknowledge or categorize Events marked as Read Only. This behavior requires custom programming with the EVENTTrak controls.
<b>Message Data</b>	
Message Data	Field for attaching extra information to each Event notification LOGICTrak sends to the EVENTTrak (for this Sub-System).

## Event Definitions

Event Definitions specify what Events to track for the parent System. When an Event Definition becomes active, LOGICTrak logs an Event and, depending on the configuration, notifies the appropriate EVENTTrak nodes. If multiple Event Definitions are active, LOGICTrak uses re-evaluation and priority rules to determine which Event Definition is responsible; it records the remaining active Event Definitions as Sub Events.

Event Definitions do not appear in the MODELTrak tree view. They appear in the list view when you select either the parent System or the parent Event Definition Group in the tree view.

## Properties

<b>General</b>	
Event Definition Type	An Event Definition Type categorizing this Event Definition (solely for organizational and reporting purposes).
Trigger Tag	Discrete Tag which triggers this Event Definition. When True, the Event Definition is active; when False, the Event Definition is not active (although other Event Definitions in the System may be active).
Ignore System Suppression	When checked, the suppression Tag functionality configured for the parent System is ignored.
Trigger on False	When checked, the Event Definition is triggered when the Trigger Tag is set to False (rather than the default behavior of triggering when True).
Default Event	The default Event Category specifies the category that the

Category	Event Definition's Events will receive. Depending on the Assigned Event Category Groups and EVENTTrak configuration, operators can select a different Event Category when they acknowledge Events.
Priority	An Integer indicating this Event Definition's priority relative to other Event Definitions in the parent System. If two or more Event Definitions become active within the same scan, LOGICTrak will assign the Event to the Event Definition with the lowest priority. LOGICTrak will record the remaining active Event Definitions as Sub-Events. Refer to the section on Event Definition Priority in Advanced Topics for more information.
OEE Event Type	The type of OEE loss this Event Definitions represents. Refer to the OEE section for more information on the OEE equation. Valid options are N/A, Performance Loss, Availability Loss, and System Not Scheduled. If you are not tracking OEE for the parent System, select N/A.
Availability Loss Delay (Seconds)	For Performance Loss and System Not Scheduled OEE Event Type settings, this value specifies the number of seconds before Events will automatically change to Availability Loss. Note that when the OEE Event Type is System Not Scheduled, LOGICTrak will split the active Event and create a new one with an OEE Event Type of Availability Loss, whereas for Performance Loss Event Definitions, LOGICTrak will reclassify entire Event as Availability Loss.
Re-Evaluate System Event on Start	When checked, this option directs LOGICTrak to re-evaluate the active Event Definitions when the Event Definitions becomes active (starts) to determine whether or not the Event Definition (or another) has lower priority. If LOGICTrak finds an active Event Definition with a lower priority, it will split the current Event and assign the lower priority Event Definition as the root cause of the new Event.
Re-Evaluate System Event on End	When checked, this option directs LOGICTrak to re-evaluate the active Event Definitions when the Event Definition becomes inactive (ends) to determine whether or not to split the current Event when another Event Definition has lower priority than the one currently assigned as the root cause of the current Event. If this Event Definition is currently the root cause of the current Event and another Event Definition is still active, LOGICTrak will always split the current Event.
Show in Reports	Determines if this entity is displayed in report pickers.
<b>EVENTTrak</b>	
EVENTTrak Nodes	List of EVENTTrak nodes that will receive Event notifications when this Event Definition becomes active. Refer to the section on Event Definition for more information.
Synchronize	Copies the selected EVENTTrak's settings to every other EVENTTrak in the list.
<b>Categories</b>	
Available Event Category Groups	List of Event Category Groups available for assignment to this Event Definition. Select one from the drop-down list and then click Add to add it to the list of Assigned Event Category Groups.

Assigned Event Category Groups	List of Event Category Groups already assigned to this Event Definition. You can reorder these using the up and down arrows and remove items using the Remove button.
<b>VTRs</b>	
VTRs	List of VTR Definitions assigned to this Event Definition. Refer to the section on Event Definition VTRs for more information.
<b>Tags</b>	
Tags	List of related Tags to this Event Definition. The relationship can be used in certain reports to automatically list or trend Tags based on the selected Event Definition.
<b>Advanced</b>	
Script Class Name	Specifies the class name that will contain code that will run for this Event Definition. The class name specified here must be implemented as an Advanced Script Library in the Administration section of MODELTrak.
MTBF Type	Determines how events recorded for the Event Definition effect MTBF calculations. "Failure" indicates that this Event Definition represents an equipment failure (MTBF is the mean time between equipment failures). "Non-Failure" indicates an event that should not be counted as an equipment failure for MTBF. During a "Non-Failure", the System is considered to be running from the standpoint of the MTBF calculation. "Excluded" indicates that this Event Definitions represents a period of time that should be completely disregarded by the MTBF calculation (for example time that the plant is closed should not count as total time in the MTBF calculation).
Duration (Seconds)	The expected average duration for Events, used primarily for custom manual event entry.
Sub Priority	Integer setting for this Event Definition's Sub-Priority, which may be necessary in rare cases when Priority alone is not sufficient to specify correct Event splitting behavior. Refer to the section on Sub-Priority in Advanced Topics for more information on Sub-Priority.
Isolation Type	This setting allows Event Definitions to be configured to record regardless of if there are other Event Definitions being active. The default setting is Single Active Event which is the standard TrakSYS behavior (one Event Definition active at a time based on Priority. Setting the Isolation Type to Independently Active will cause the Event Definition to be recorded (an Event record created) whenever its Trigger Tag is true, regardless of the state of the other Event Definitions in the System. In this mode, the Priority settings are ignored.

### Event Definition EVENTTrak Node Definitions

Represent EVENTTrak nodes assigned to an Event Definition, with additional settings for operator interaction. EVENTTrak nodes cannot be assigned at the Event Definition level if they are already assigned at the System or Sub-System level.

#### Properties

<b>General</b>	
EVENTTrak Node	The EVENTTrak node assigned to this Event Definition.

Display Delay (Seconds)	Number of seconds after an Event begins before it appears in EVENTTrak.
Acknowledge Delay (Minutes)	Number of minutes operators have to acknowledge an Event before EVENTTrak auto-acknowledges it.
Allow Acknowledge While Active	When checked, allows operators to acknowledge and categorize Events while they are still active. This behavior requires custom programming with the EVENTTrak controls.
Force Acknowledge On Event End	When checked, EVENTTrak will automatically acknowledge Events as soon as they end.
Require Event Category	When checked, this requires operators to select an Event Category in order to acknowledge an Event. This behavior requires custom programming with the EVENTTrak controls.
Read Only	When checked, this Event Definition's Events will appear read-only in the EVENTTrak. Operators will not be able to acknowledge or categorize Events marked as Read Only. This behavior requires custom programming with the EVENTTrak controls.
Do Not Send	Typically, the EVENTTrak settings set at the System level are inherited down to the Event Definition level. When this setting is checked, it overrides that behavior and prevents messages from being sent for to this node for this Event Definition.
<b>Message Data</b>	
Message Data	Field for attaching extra information to each Event notification LOGICTrak sends to the EVENTTrak (for this Event Definition).

### Event Definition VTRs

Event Definition VTR (Value-To-Record) Definitions specify supplementary Tag Values to record when an Event starts or ends.

#### Properties

<b>General</b>	
Record Tag	Tag whose value LOGICTrak should record when an Event occurs.
Event Edge	The time at which LOGICTrak should record the VTR. Select "Event Start" for LOGICTrak to record the VTR when an Event starts, or select "Event End" for LOGICTrak to record the VTR when an Event ends.

### Event Definition Tag Relationships

Event Definition Tag Relationships provide a way to associate Tags with an Event Definition. There is no data collected or stored automatically as a result of configuring Tag Relationships, they exist for extensibility purposes only (for instance creating a report that trends a set of Tags based on the selected Event Definition).

#### Properties

<b>General</b>	
Relate Tag	The Tag that will be associated with this

	entity.
Abbreviation	A shorter name or abbreviation for the related Tag.

## Task Definitions

Task Definitions allow the specification of a condition that will trigger a Task record to be created in the database. Tasks are records that require user input such as notes or specific values to be entered. The exact details of the information required by a specific Task are configured in Task Form Items.

### Properties

<b>General</b>	
Task Definition Type	A Task Definition Type categorizing this Task Definition (solely for organizational and reporting purposes).
Trigger Tag	Discrete Tag which triggers this Task Definition. When the Tag transitions from False to True, the Task Definition causes the creation of a new Task record in the database.
Task Completion (Seconds)	This specifies the number of seconds that an end user has to make the data entry associated with the Task. It can be used to enforce rules in the user interface that is created for Task entry.
<b>Task Form Item Groups</b>	
Available Task Form Item Groups	List of available Task Form Item Groups you may assign to this Task Definition.
Assigned Task Form Item Groups	List of Task Form Item Groups that are assigned to this Task Definition.
<b>VTRs</b>	
VTRs	List of VTR Definitions assigned to this Task Definition. Refer to the section on Task Definition VTRs for more information.
<b>Advanced</b>	
Script Class Name	Specifies the class name that will contain code that will run for this Task Definition. The class name specified here must be implemented as an Advanced Script Library in the Administration section of MODELTrak.

## Task Definition VTRs

Task Definition VTR (Value-To-Record) Definitions specify supplementary Tag Values to record when a Task record is created. Task records are created when the associated Task Definition's Trigger Tag transitions from False to True.

### Properties

<b>General</b>	
Record Tag	Tag whose value LOGICTrak should record when a Task record is created.

## Task Form Item Groups

A Task Form Item Group contains a set of Task Form Items. One or more of these Groups can be referenced by a Task Definition. When a new Task record is created for a Task Definition, placeholders for data entry are created for all of the Task Form Items in all of the referenced Groups.

### Properties

General	
Key	A unique string that identifies the Task Form Item Group.

## Task Form Items

A Task Form Items represents a single piece of data that is to be recorded with a Task record. Task Form Items are grouped using Task Form Item Groups and then assigned to Task Definitions.

### Properties

General	
Key	A unique string that identifies this Task Form Item.
Data Type	Indicates that Data Type for the value specified for the Task Form Item.
Default Value	Specifies the default value that may appear in the entry field for this Task Form Item when displayed for input.
Enum Values	Lists the possible Task Form Item Enums (name/value pairs) for the Task Form Item.

## Task Form Item Enums

Task Form Item Enums represent possible value selections for a Task Form Item.

### Properties

General	
Value	Specifies the value that will be stored for the Task Form Item if this choice is selected in a data entry user interface.

## Function Definitions

A Function Definition defines a potential Batch System operation (it is only available under Batch Systems). When defining Batch Recipes, each step in the Recipe must be assigned to a Function Definition. A Recipe step is automatically started when its associated Function Definition becomes active based on its Trigger Tag logic.

### Properties

General	
Function Definition Type	A Function Definition Type categorizing this Function Definition (solely for organizational and reporting purposes).
Trigger Tag	A Discrete Tag which triggers this Function Definition. When True, the Function Definition is active; when False, the Function Definition is not active.
Default Overage	Specifies an optional Step Category that is assigned by default



Step Category	to any time the Function exceeds its planned duration (as defined in the Recipe Step Definition)
Overage OEE Step Type	Specifies an optional OEE Step Type that is assigned by default to any time the Function exceeds its planned duration (as defined in the Recipe Step Definition).
<b>Categories</b>	
Available Step Category Groups	List of Step Category Groups available for assignment to this Function Definition. Select one from the drop-down list and then click Add to add it to the list of Assigned Step Category Groups.
Assigned Step Category Groups	List of Step Category Groups already assigned to this Function Definition. You can reorder these using the up and down arrows and remove items using the Remove button.
<b>Parameters</b>	
Parameters	List of Parameters that can be assigned specific values when this Function Definition is assigned to a Recipe Step.
<b>Tags</b>	
Tags	List of Tags related to this System. These relationships may be used in reporting to list or trend Tags based on the selected Function Definition.
<b>Advanced</b>	
Script Class Name	Specifies the class name that will contain code that will run for this Function Definition. The class name specified here must be implemented as an Advanced Script Library in the Administration section of MODELTrak.

## Function Definition Tag Relationships

Function Definition Tag Relationships provide a way to associate Tags with a Function Definition. There is no data collected or stored automatically as a result of configuring Tag Relationships, they exist for extensibility purposes only (for instance creating a report that trends a set of Tags based on the selected Function Definition).

### Properties

<b>General</b>	
Relate Tag	The Tag that will be associated with this entity.
Abbreviation	A shorter name or abbreviation for the related Tag.

## Function Definition Parameter Definitions

Parameter Definitions allow the specification of a value that is to be recorded with the Batch or specific Batch Step. The actual value that is recorded is based on configuration set at the Recipe that is associated with the Batch at runtime.

### Properties

<b>General</b>	
Parameter Definition Type	A Parameter Definition Type categorizing this Parameter Definition (solely for organizational and reporting purposes).
Data Type	The Parameter's data type. Select

	String, Integer, Float, or Discrete.
Default Value	The initial value for this Parameter that appears in the Recipe configuration screens.
Parameter Tag	If specified, the Parameter value specified in the Recipe is written to this Tag. This Tag must be an OPC or Virtual Tag. OPC Tags can only be written to if the license allows it.
Minimum Value	The minimum value that can be configured in the Recipe for this Parameter.
Maximum Value	The maximum value that can be configured in the Recipe for this Parameter.
Exception Value	The value that will be assigned to this Parameter when a Batch Step gets triggered in an unpredicted order.

## Event Categories

Event Categories specify predefined values for operators to categorize Events reported to the EVENTTrak. You may assign Event Categories to different hierarchical levels and create an Event Category Scheme to define each level.

### Properties

<b>General</b>	
Event Category Type	An Event Category Type categorizing this Event Category (solely for organizational and reporting purposes).
Event Code	This Event Category's Event Code or "super reason," which relates similar Event Categories.
Inherit Event Code from Parent	When checked, specifies that this Event Category should inherit its Event Code from its parent Event Category. This setting does not apply to top-level Event Categories.
OEE Type	Type of OEE loss this Event Category represents. This setting allows operators to override the Event Definition's OEE Event Type setting by selecting this Event Category from the EVENTTrak. Select N/A to keep the Event Definition's OEE Event Type setting.
Inherit OEE Type from Parent	When checked, specifies that this Event Category should inherit its OEE Type setting from its parent Event Category. This setting does not apply to top-level Event Categories.
Selectable	When checked, specifies that operators may select this Event Category in

	EVENTTrak. In hierarchical category schemes, you may want to prevent operators from selecting higher or top-level Event Categories, forcing them to choose lower-level, more detailed Event Categories.
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Event Category Groups

Event Category Groups are sets of Event Categories that you can assign to Systems and Event Definitions for operator selection using EVENTTrak.

### Properties

General	
Scope	Setting for whether this Event Category Group is normal or Global. Global Event Categories are available on every EVENTTrak for every Event Definition; you do not need to explicitly assign Global Event Categories to Event Definitions or Systems. Select Global First if you would like the Global Event Category to appear at the top of the Event Category selection list in EVENTTrak, or select Global Last if you would like it to appear at the bottom of the list. If the Scope is Normal, you must assign Event Categories for them to be available in EVENTTrak.
Event Category Scheme	The scheme defining each Event Category level. Event Category Schemes associate names with each level of categorization, such as Reasons and Sub-Reasons.

## Event Category Schemes

Event Category Schemes allow you to define hierarchical levels associated with Event Categories. For example, the first level of Event Categories may represent Reasons, whereas the second level may represent Sub-Reasons. Using Event Categories with levels and an Event Category Scheme, you can define a complete hierarchy which EVENTTrak will present to operators as a tree view. The reports also use Event Category Schemes to properly label Event Categories levels.

### Event Category Scheme Levels

An Event Category Scheme Level is a label for a level of hierarchy in the Event Category Tree View structure. These labels are used for reference in the MODELTrak menus and the WEBTrak reports.

## Event Codes

Event Codes help organize similar or related Event Categories. These may be useful for relating Event Categories to an existing business categorization scheme. Once you have related Event Categories to Event Codes, EVENTTrak will automatically record the corresponding Event Code when operators categorize an Event.

### Properties

General	
Event Code Type	An Event Code Type categorizing this Event Code (solely for organizational and reporting purposes).

Code	Custom alphanumeric Code string.
------	----------------------------------

## KPI Calculations

LOGICTrak uses KPI Calculations to compute and record KPI information for Systems. TrakSYS requires a theoretical production rate and special Tags called Counters to accurately track KPI. In order to provide updated KPI information throughout the day, LOGICTrak records KPI values into Intervals, usually between 30 and 60 minutes in duration. Once an Interval is complete, you can use the reports to view its calculated KPI.

KPI Calculations do not appear in the MODELTrak tree view. You must select the parent System in order to view its KPI Calculations in the list view.

## Properties

<b>General</b>	
Interval Duration (Minutes)	The frequency at which LOGICTrak will record KPI information to the database. Smaller values make KPI information available to the reports sooner, but create more Intervals, consume more database space, may reduce performance, and make it more difficult to manually adjust KPI data. Typical values for this setting are between 30 and 60 minutes.
Derived Input	Which of the three Counter inputs (Good, Bad or Total) LOGICTrak should automatically calculate based on the other two. Since $\text{Total} = \text{Good} + \text{Bad}$ , you need only specify two of the equation's inputs to calculate the third.
Calculation Units	Label for the Calculation's units (i.e. blisters, tablets, wallets, etc.). This setting is for informational purposes only.
Theoretical Rate (Total Units / Minute)	The theoretical number of units the monitored equipment can produce in 1 minute. This value may be available from the equipment manufacturer. To prevent possible inaccuracies in the KPI data, this setting should be a maximum; if the actual production rate exceeds this value your KPI and Performance may exceed 100%.
Target Rate (Good Units)	The rate at which production should occur to meet goals. This is can be used for KPIs such as Takt or as a variable target line for efficiency. This value can be specified as a Constant, a Tag, or Calculated (based on the Theoretical Rate above).
Target Rate Units Type	Specifies that the Target Rate is expressed in terms of Seconds per Units or Units per Minute.
Show in Reports	Determines if this entity is displayed in report pickers.
<b>Total Counters</b>	
Input Units	Label for the Total Input's units. This setting is for informational purposes only.
Input Units to Calculation Units Multiplier	Floating-point multiplier to convert the Total Input's Units to Calculation Units. If the Units are identical, simply specify 1.0. If all of your Total Counters use the same Units, you should use this setting to specify the conversion to Calculation Units rather than setting identical conversions for each Total Counter.
Flow On Input Units Count	Number of Input Units that must pass through the Input within the time specified by Flow On Period before the Input is Flowing.

Flow On Period (Seconds)	A period of seconds during which the number of units specified by Flow On Input Units Count must pass through the Input before it is Flowing. Note that the Input can begin Flowing sooner if it reaches Flow On Input Units Count in less time than the Flow On Period.
Flow Off Delay (Seconds)	Delay after the counters stop incrementing before this Input stops Flowing.
<b>Good Counters</b>	
Input Units	Label for the Good Input's units. This setting is for informational purposes only.
Input Units to Calculation Units Multiplier	Floating-point multiplier to convert the Good Input's Units to Calculation Units. If the Units are identical, simply specify 1.0. If all of your Good Counters use the same Units, you should use this setting to specify the conversion to Calculation Units rather than setting identical conversions for each Good Counter.
Flow On Input Units Count	Number of Input Units that must pass through the Input within the time specified by Flow On Period before the Input is Flowing.
Flow On Period (Seconds)	A period of seconds during which the number of units specified by Flow On Input Units Count must pass through the Input before it is Flowing. Note that the Input can begin Flowing sooner if it reaches Flow On Input Units Count in less time than the Flow On Period.
Flow Off Delay (Seconds)	Delay after the counters stop incrementing before this Input stops Flowing.
<b>Bad Counters</b>	
Input Units	Label for the Good Input's units. This setting is for informational purposes only.
Input Units to Calculation Units Multiplier	Floating-point multiplier to convert the Bad Input's Units to Calculation Units. If the Units are identical, simply specify 1.0. If all of your Bad Counters use the same Units, you should use this setting to specify the conversion to Calculation Units rather than setting identical conversions for each Bad Counter.
Flow On Input Units Count	Number of Input Units that must pass through the Input within the time specified by Flow On Period before the Input is Flowing.
Flow On Period (Seconds)	A period of seconds during which the number of units specified by Flow On Input Units Count must pass through the Input before it is Flowing. Note that the Input can begin Flowing sooner if it reaches Flow On Input Units Count in less time than the Flow On Period.
Flow Off Delay (Seconds)	Delay after the counters stop incrementing before this Input stops Flowing.
<b>Capture</b>	
Capture Scheme	The KPI Capture Schemes defining each of the Interval Capture Columns. The Scheme defines the Name, Data Type, Edge Type, and whether or not the Capture Tag should split the Interval when it changes.
Capture Scheme Columns	List of the KPI Capture Scheme Columns and their assigned KPI Capture Tags. The Columns are defined by the selected Capture Scheme. Double-click on a Column to configure a KPI

	Capture Tag.
<b>VTRs</b>	
VTRs	List of KPI VTR Definitions assigned to this Calculation.
<b>Targets</b>	
Baseline OEE %	Informational setting displayed in the reports for tracking your improvement in OEE. This is typically your starting OEE when you first implement TrakSYS.
Baseline Availability %	Informational setting displayed in the reports for tracking your improvement in Availability. This is typically your starting Availability when you first implement TrakSYS.
Baseline Performance %	Informational setting displayed in the reports for tracking your improvement in Performance. This is typically your starting Performance when you first implement TrakSYS.
Baseline Quality %	Informational setting displayed in the reports for tracking your improvement in Quality. This is typically your starting Quality when you first implement TrakSYS.
Baseline TEEP %	Informational setting displayed in the reports for tracking your improvement in TEEP. This is typically your starting TEEP when you first implement TrakSYS.
Target OEE %	Informational setting displayed in the reports for tracking your progress toward a target OEE. For reference, World Class OEE is 85% or greater.
Target Availability %	Informational setting displayed in the reports for tracking your progress toward a target Availability. For reference, World Class Availability is 90% or greater.
Target Performance %	Informational setting displayed in the reports for tracking your progress toward a target Performance. For reference, World Class Performance is 95% or greater.
Target Quality %	Informational setting displayed in the reports for tracking your progress toward a target Quality. For reference, World Class Quality is 99.9% or greater.
Target TEEP %	Informational setting displayed in the reports for tracking your progress toward a target TEEP.
<b>Advanced</b>	
Script Class Name	Specifies the class name that will contain code that will run for this KPI Calculation. The class name specified here must be implemented as an Advanced Script Library in the Administration section of MODELTrak
Display Units	Label for the Display Units in the reports.
Calculation Units to Display Units Divisor	Numeric divisor to convert the Calculation Units to Display Units.
Interval Date	What date LOGICTrak should assign to KPI Intervals as they are recorded. Select "Interval Start Date" to use the date as the Interval starts. Select "Interval End Date" to use the date as the Interval ends. Select "Interval Shift Date" to use the date assigned to the current Shift for the parent System.
Suppress Counts	When checked, KPI Intervals that are completely covered by a combination of System Not Scheduled and Availability Loss events, return production counts of 0 for the purposes of reporting. This setting helps prevent skewed OEE data for Intervals that contain rouge production counts during events

	like maintenance or product changeover. Note that production counts are always recorded to the database, this setting controls whether or not they are suppressed in reports.
Update Interval (Minutes)	Determines if and how often LOGICTrak updates the current KPI Interval in the database. When the value of this setting is "None", the KPI Interval is not updated with any data until it has ended. When the value is "Automatic", LOGICTrak will automatically determine the optimal update rate (~10% of the Interval Duration).
Interval Midnight Offset (Minutes)	Offset from 12:00 AM to use as a user-definable midnight for synchronizing KPI Intervals. For example, if you set the Interval Midnight Offset to +30 minutes, and then set Interval Duration to 60 minutes, LOGICTrak will write KPI Intervals as 12:30 AM, 1:30 AM, 2:30 AM, and etc, even if Capture Tag or Shift changes force Interval splits between those times. A common use of this setting is to synchronize KPI Intervals with the Shift Schedule.
Enable Notification	Specifies that the KPI Calculation should report its value changes to ALERTTrak for subscriptions. This property is automatically set when a KPI Calculation is referenced by an ALERTTrak subscription.

## KPI Counters

KPI Counters track Total, Good, and Bad production counts. Rather than using the actual values of the Counter Tags, LOGICTrak monitors the changes in their values to more accurately track their movement.

Flow provides a simple mechanism to track when a KPI Counter (and the line) is running. To prevent small, intermittent counter movement from activating a Counter's Flow status, LOGICTrak uses a Flow On ratio to determine when to set the Counter's state to Flowing. For the Flow to activate, the Counter must increment by the Flow On Counter Units Count within the time specified by Flow On Period. Once a Counter is Flowing, it does not reset until the Counter stop incrementing for the time specified by Flow Off Delay.

For example, if the Flow On Counter Units Count is 5, the Flow On Period is 10 seconds, and the Flow Off Delay is 15 seconds, the KPI Counter must increment by 5 (Counter Units) within a 10 second span before Flow activates. Once Flowing, the Flow will remain active until the Counter stops incrementing for 15 consecutive seconds.

The Flow status of a KPI Counter is available as a State Tag. The State Tag Properties dialog in MODELTrak allows you to select the desired KPI Counter and state information.

## Properties

<b>General</b>	
Counter	Tag or constant providing the Counter value.
Counter Units to Input Units Multiplier	Floating-point multiplier to convert Counter Units to Input Units. If the Units are identical, simply specify 1.0.
Maximum	The maximum number of (Counter) units the Counter may

Increment per Scan	increment within a single scan. Set this to the maximum value the Counter can realistically increment to filter out unexpected large jumps in Counter values. Cycling the power on PLCs or uploading new programs may cause erratic Counter behavior that can severely impact OEE. Note that any time a Counter moves backward, LOGICTrak considers it a Rollover event (reset to 0) and uses this same setting to determine whether or not to discard the change in Counter value (New Value – 0).
KPI Count Category	Specifies the KPI Count Category that will be assigned to Interval Count records recorded for this KPI Counter.
<b>Flow</b>	
Flow On Counter Units Count	Number of Counter Units that must pass through the Counter within the time specified by Flow On Period before the Counter is Flowing.
Flow On Period (Seconds)	A period of seconds during which the number of units specified by Flow On Counter Units Count must pass through the Counter before it is Flowing. Note that the Counter can begin Flowing sooner if it reaches Flow On Counter Units Count in less time than the Flow On Period.
Flow Off Delay (Seconds)	Delay after the Counter stops incrementing before it stops Flowing.
<b>Advanced</b>	
Rollover	<p>Maximum value at which the Counter rolls over to zero. Depending on the scan rate, LOGICTrak may miss the maximum Counter value on some high-speed lines. If Rollover always occurs at a specific value, you can use this setting to ensure LOGICTrak interprets the Rollover correctly. Note that the Maximum Increment per Scan setting overrides this setting, so if the Rollover is incorrect (i.e. too high) and the adjusted Counter change exceeds the Maximum Increment per Scan, LOGICTrak will discard the change altogether.</p> <p>For example, if a particularly fast Counter's Rollover occurs at 10, but LOGICTrak only sees the values 9 and then 1, it will assume the Rollover occurred at 9 and interpret the change as +2. However, if you explicitly set Rollover to 10, LOGICTrak will correctly adjust this to +3 to account for the missed Counter change from 9 to 10. Note that if the Maximum Increment per Scan setting is less than 3, LOGICTrak will discard the Counter change. Therefore, it is important that Rollover be set correctly; otherwise, an incorrect value may make the data less accurate than not specifying Rollover at all.</p>
Record Detail	When True, LOGICTrak will record detailed tracking data every time the Counter changes. Use this setting to troubleshoot problematic Counters. You can use the KPI Counter Detail administrative report to view these values.

## KPI VTR Definitions

KPI VTR (Value-To-Record) Definitions specify supplementary Tag Values to record when a KPI Interval starts or ends.



## Properties

General	
Record Tag	Tag whose value LOGICTrak should record every KPI Interval.
Interval Edge	The time at which LOGICTrak should record the VTR. Select "Interval Start" for LOGICTrak to record the VTR when an Interval starts, or select "Interval End" for LOGICTrak to record the VTR when an Interval ends.

## KPI Capture Schemes

A KPI Capture Scheme defines the Capture Columns associated with a KPI Calculation. Once you assign a Capture Tag to a Capture Scheme Column in your KPI Calculation, LOGICTrak will record the values of the Capture Tag as supplementary data in a matching column for each Interval. Capture Tags work similarly to VTRs, but the reports allow you to report by Capture Columns rather than merely view the Captured values in a detail view. Common Capture Columns are Job, Batch, and Lot Number.

## KPI Capture Scheme Columns

### Properties

General	
Data Type	The data type for the Column. Select String, Integer, Float, or Discrete.
Capture on Interval Edge	The time at which LOGICTrak should capture the associated Capture Tag's value. Select "Interval Start" for LOGICTrak to record the Capture Tag when an Interval Starts, or select "Interval End" for LOGICTrak to record the Capture Tag when an Interval ends.
Split Interval on Value Change	When checked, LOGICTrak will split the current Interval whenever the associated Capture Tag's value changes. This setting is necessary if you wish to report by this Column; without it, there would be no record of when the Capture Tag changed. Note that using this setting does not interfere with the Interval Midnight Offset for the Calculation.

## KPI Capture Tags

### Properties

General	
Capture Tag	The Tag for LOGICTrak to record for the corresponding Capture Scheme Column.
Data Type	The Data Type of the KPI Capture Scheme Column. The Column and Capture Tag's Data Types must match.
Interval Edge Type	The Interval Edge Type of the KPI Capture Scheme Column.
Split Interval on Value Change	The Split Interval on Value Change setting of the KPI Capture Scheme Column.

## KPI Calculation Views

KPI Calculation Views allow you to arbitrarily group and order sets of KPI Calculations for use in MODELTrak.

### Properties

General	
Available KPI Calculations	List of KPI Calculations available to add to this view. Select a KPI Calculation from this list and then click Add to add it to the list of Assigned KPI Calculations.
Assigned KPI Calculations	List of KPI Calculations assigned to this KPI Calculation View. KPI Calculations in this list will appear in the view.

## KPI Count Categories

KPI Count Categories are user defined descriptors that can be applied to KPI Counters to add an additional level of report grouping when breaking down production count for KPI Calculations. The most common use of a KPI Count Category is with multiple Bad production counters. KPI Count Categories can be used to group certain types of Bad production counters together for different perspectives during data analysis. KPI Count Categories are assigned for use in the KPI Counter property page.

## Sample Categories

Sample Categories are reasons that can be selected as causes for bad samples. The most common use of a Sample Category is when the Sample Definition is defined as Percent mode. In this mode, a fixed number of items are measured and the number of bad samples is recorded for different reasons (Sample Categories). The Sample Categories defined here are associated with the sample data using the APIs in EVENTTrak or Advanced Script Tags.

## Journal Categories

Journal Categories are predefined types that can be applied to IMPROVETrak Journal entries. A Sample Category/Journal Category defined here is listed in the Journal entry forms.

## Step Categories

Step Categories are reasons that can be applied to Batch Steps that exceed their planned duration that explain the overage. A Step Category is analogous to Event Category for Events.

## Step Category Groups

Step Category Groups are sets of Step Categories that can be assigned to Batch Sub-Systems and Function Definitions for user selection in EVENTTrak.

## Step Category Schemes

Step Category Schemes allow the definition of hierarchical levels associated with Step Categories. For example, the first level of Step Categories may represent Reasons, whereas the second level may represent Sub-Reasons. Using Step Categories with levels and a Step Category Scheme, you can define a complete

hierarchy which EVENTTrak will present to operators as a tree view. The reports also use Step Category Schemes to properly label Step Category levels.

### **Step Category Scheme Levels**

A Step Category Scheme Level is a label for a level of hierarchy in the Step Category Tree View structure. These labels are used for reference in the MODELTrak menus and the WEBTrak reports.

### **Step Codes**

Step Codes help organize similar or related Step Categories. These are useful for relating Step Categories to an existing business categorization scheme. Once Step Categories are related to Event Codes, EVENTTrak will automatically record the corresponding Step Code when operators categorize a Batch Step.

### **Batch Quality Loss Categories**

Batch Quality Loss Categories are reasons that can be assigned to bad production as the result of a Batch. A Batch Quality Loss Category is similar to Event Category for Events.

### **Products**

Products represent what the equipment being monitored produces, and specify additional attributes such as theoretical rates and multipliers for KPI. Each Product has a unique identifying Product Code, which is often a SKU number.

Product Schemes define the Product Attributes associated with your Products, such as size, finish, and KPI theoretical rate. Products within the same Product Set must have the same Product Scheme.

Product Sets are complete lists of Products applicable to one or more Systems. If you have many Products in your Product Sets, you may wish to further divide your Sets into Product Groups for better organization.

Once you define your Product Sets and assign them to the appropriate Systems, you “map” your Product Sets to Product Tags using Product Maps. Product Maps relate a single Product Tag to the Product Codes of your Product Sets.

### **Product Schemes**

Product Schemes define the available Attributes for Products. A Product Scheme is a set of 20 Product Attributes. Empty Attributes are created automatically when a Product Scheme is created.

### **Product Attributes**

A Product Scheme consists of a set of Product Attributes. Each Attribute defines additional information for Products. You may define up to 20 Product Attributes for a Product Scheme.

### **Properties**

<b>General</b>	
Data Type	The Data Type of this Product Attribute. The corresponding Tag in the Product Map must have the same Data Type.

## Product Maps

A Product Map maps values for Products to the Attributes defined by their Product Schemes. Each Product Map contains an Item for every Attribute in the corresponding Product Scheme.

### Properties

General	
Product Set	The set of Products this Map applies to.
Product CodeTag	The Tag representing the running Product. LOGICTrak will compare the value of this Tag to the Product Codes in the Product Set to load the attributes for the current Product.
LOGICTrak	The LOGICTrak Instance that this Product Map will be loaded by (a map can only be loaded by one LOGICTrak). This property is displayed only when there is more than one LOGICTrak Instance configured in MODELTrak.

## Product Map Items

Product Map Items map individual Product Attributes to output Virtual Tags where LOGICTrak stores the properties of the current Product. In most cases, you will have a set of Virtual Tags for each physical production line; other configuration entities, such as System Capture Tags and KPI Calculations, typically reference these same Tags to obtain runtime information related to the current Product.

### Properties

General	
Map Tag	The destination Tag where LOGICTrak will write the value of the corresponding Product Attribute.

## Product Sets

Product Sets list the available Products for a System and designate a Product Scheme. They may be flat lists of Products or contain hierarchies of Products arranged into Product Groups.

### Properties

General	
Product Scheme	The Product Scheme this set of Products uses, which defines the list of Product Attributes common to this Set.
Default Product	Fallback Product LOGICTrak should use when it fails to find a matching Product in the Product Set for the Product Map's Product Code Tag value.

## Recipes

A Recipe is a formula or a set of production directions for a specific Product on a specific Batch System. Recipes include many details and child entities.

## Properties

<b>General</b>	
Recipe Type	A Recipe Type categorizing this Recipe (solely for organizational and reporting purposes).
System	The System that this Recipe is configured to run on. This lists only the Batch Systems.
Product	The Product that this Recipe is configured for. This lists only the Products in the parent Product Set.
Planned Number of Batches	The standard number of Batches that are run when this Recipe is used. This is a default and can be overridden in the actual run time application.
Planned Batch Size	The standard size of each Batch when this Recipe is used. This is a default and can be overridden in the actual run time application.
Planned Batch Size Units	The textual description of the units that are being produced in each Batch. This value is used primarily for display and reporting.
Planned Batch Duration (Minutes)	The overall planned length of the Batch. This is specified explicitly rather than being calculated by the planned durations of the Recipe Steps because of the potential for Recipe Steps to overlap each other in complex ways.
Duration By Batch Size	This table lists the Planned Batch Duration values as they vary based on the size of the Batch being run. If the Batch size is not found in the lookup table, then the Planned Batch Duration configured above will be used.
Edit Recipe Step Definitions	This launches a dialog that allows the specification of the Recipe Steps.
<b>Parameters</b>	
Parameters	This lists the Parameters that are configured for the System that is selected above. The Parameter values for each row can be specified by double clicking and accessing the Parameter Properties dialog.

## Recipe Step Definitions

A Recipe is made up of one or more Recipe Step Definitions (RSD). A Recipe Step Definition is the execution of one of the Batch System's Function Definitions for a specified duration. While Recipe Step Definitions for the overall Batch System can be defined to execute in parallel, only one Recipe Step Definition per Batch Sub-System can be configured to be active at one time.

Recipe Step Definitions are configured with a specific Start and End Sequence. The Sequence number is an arbitrary integer that represents the order in which the RSD starts and ends in relation to the other RSDs for the Recipe. When setting Sequence values, the following general rules should be observed.

- If Step A is set to end on Sequence 4 and Step B is set to start on Sequence 4, then Step B cannot start until Step A is complete.
- If Step A is set to end on Sequence 3 and Step B is set to start on Sequence 2, then Step B can start before or after Step A is complete.
- If Step A and Step B are set to start on Sequence 1 then the steps are parallel and they can start independently of each other.

- If Step A starts and ends on Sequence 1 and Step B starts and ends on Sequence 2, then Step B cannot start until Step A is complete.
- If Step A is set to start on Sequence 1 and end on Sequence 3, and Step B is set to Start on Sequence 2, Step B cannot start until Step A has started.

### Properties

<b>General</b>	
Function Definition	The Function Definition that is run when this RSD is active.
Recipe Step Definition Type	A Recipe Step Definition Type categorizing this Recipe (solely for organizational and reporting purposes).
Start Sequence	The Sequence that this RSD is set to start in.
End Sequence	The Sequence that this RSD is set to end in.
Planned Duration Seconds	The target duration for the RSD.
Duration by Batch Size	This table lists the Planned Duration Seconds values as they vary based on the size of the Batch being run. If the Batch size is not found in the lookup table, then the Planned Duration Seconds configured above will be used.
<b>Parameters</b>	
Parameters	This lists the Parameters that are configured for the Function Definition that is selected above. The Parameter values for each row can be specified by double clicking and accessing the Parameter Properties dialog.
<b>Materials</b>	
Materials	This lists the Materials and quantities that are consumed by the Batch process at this specific Recipe Step Definition.

### Recipe Duration Ranges

A Recipe Duration Range helps specify the planned duration of the overall Recipe in relation to the Batch's planned size.

### Properties

<b>General</b>	
Batch Size Start Units	The lower end of the size range for this entry.
Batch Size End Units	The upper end of the size range for this entry.
Duration Seconds	If the Batch's planned size falls between the Start and End Units specified above, the planned duration of the Batch will be set with this value.

## Recipe Step Definition Duration Ranges

A Recipe Duration Range helps specify the planned duration of the overall Recipe in relation to the Batch's planned size.

### Properties

General	
Batch Size Start Units	The lower end of the size range for this entry.
Batch Size End Units	The upper end of the size range for this entry.
Duration Seconds	If the Batch's planned size falls between the Start and End Units specified above, the planned duration of the Batch will be set with this value.

## Recipe Step Definition Parameters

Recipe Step Definition Parameters allow Recipe specific values to be assigned to pre-defined Function Definition Parameters defined in the equipment model (Areas and Systems). Values for RSD Parameters are assigned to the Tag defined in the Function Definition Parameter when the RSD starts.

### Properties

General	
Parameter Definition	This is the name of the Parameter Definition as defined for the Function Definition that the current RSD is based on.
Value	This is the value that is assigned to the Function Definition Parameter Tag when the RSD starts.
Value by Batch Size	This table lists the values as they vary based on the size of the Batch being run. If the Batch size range is not found in the lookup table, then the Value configured above will be used.

## Recipe Step Definition Parameter Ranges

The RSD Parameter Range specifies a Parameter value for a specific Batch size (range).

### Properties

General	
Batch Size Start Units	The lower end of the size range for this entry.
Batch Size End Units	The upper end of the size range for this entry.
Value	The value that will be assigned for the parent Parameter if the current Batch size falls between the configured Start and End Unit range.

## Recipe Step Definition Materials

A Recipe Step Definition Material a specific amount of a defined Material that must be added for the parent Step.

### Properties

General	
Material	The Material that is required to be added for the parent Step.
Quantity	The Quantity of the defined Material to be added for the parent Step.

## Recipe Parameters

Recipe Parameters allow Recipe specific values to be assigned to pre-defined Batch System Parameters defined in the equipment model (Areas and Systems). Values for Recipe Parameters are assigned to the Tag defined in the Batch System Parameter when the Recipe starts.

### Properties

General	
Parameter Definition	This is the name of the Parameter Definition as defined for the Batch System that the current Recipe is based on.
Value	This is the value that is assigned to the Batch System Parameter Tag when the Recipe starts.
Value by Batch Size	This table lists the values as they vary based on the size of the Batch being run. If the Batch size range is not found in the lookup table, then the Value configured above will be used.

## Recipe Parameter Ranges

The Recipe Parameter Range specifies a Parameter value for a specific Batch size (range).

### Properties

General	
Batch Size Start Units	The lower end of the size range for this entry.
Batch Size End Units	The upper end of the size range for this entry.
Value	The value that will be assigned for the parent Parameter if the current Batch size falls between the configured Start and End Unit range.



## Materials

A Material represents something that is consumed or produced by the manufacturing process. Materials can be simple “raw” ingredients or more complex combinations of several other defined Materials in a specific ratio.

### Properties

<b>General</b>	
Material Type	A Material Type categorizing this Material (solely for organizational and reporting purposes).
Material Code	A string that identifies this Material. The Material Code must be unique within the entire TrakSYS configuration.
Units	A textual description of the unit of measurement for this Material.
<b>Components</b>	
Planned Size	If the Material is a compound made up of other Materials, this is an arbitrary quantity that helps contextualize the amounts of the sub-components that are listed below.
Material Component	If the Material is a compound made up of other Materials, this is a list of the sub-components and their quantities.
<b>Alternates</b>	
Material Alternate	This is an optional list of other Materials that may be substituted for this Material if it is not available for use. This list is provided for customizations and is not used within the TrakSYS standard applications.

## Material Components

A Material Component is a reference and quantity of another existing Material. They are used to construct compound Materials.

### Properties

<b>General</b>	
Material Component	The referenced sub-component Material.
Quantity	The quantity of the sub-component Material needed to make a specified amount of the parent Material.

## Material Alternates

A Material Alternate is a reference to another existing Material that may be used as an alternate if the original Material is not available.

### Properties

<b>General</b>	
Material Group	This lists the available Material Groups and acts as a filter for the Material Alternate picker below.
Material Alternate	The referenced Material that can be used as an alternate to the parent Material.

## Schedules

The LOGICTrak uses Schedules to determine active and upcoming Shifts for Systems.

### Properties

General	
Reload Interval (Seconds)	The frequency LOGICTrak should check the database for changes to the Schedule. Changes to the active Schedule do not occur until this interval elapses. The recommended setting is approximately 90 seconds.
Allow Schedule Edits (Edit Mode)	Places the Schedule into Edit Mode, which allows modifications to its Patterns. While in Edit Mode, LOGICTrak will not load changes to the Schedule, and will continue using the last Pattern loaded into memory, if one exists. This effectively disables the Reload Interval setting. The purpose of this setting is to prevent incomplete Schedule Patterns from taking effect and corrupting Shift data. Note that LOGICTrak will not start if one or more of the Schedules referenced by the configuration are in Edit Mode.

## Schedule Patterns

Schedule Patterns allow you to configure rotating Shifts for a Schedule. A Pattern can be any number of days long and contain any number of Pattern Items representing distinct Shift instances. Each Schedule can contain multiple Patterns, but only one Pattern may be active at a time. LOGICTrak chooses the active Schedule based on the one with the most recent Effective Date/Time setting.

### Properties

General	
Start Date/Time	<p>The date and time this Schedule Pattern begins. Note that this setting is solely for synchronization purposes; since Schedule Patterns rotate, this setting may be any date as long as it corresponds to one of the correct dates on which the Schedule Pattern repeats.</p> <p>For example, if your Schedule Pattern is a week long, only the Start Date/Time's day of week needs to match. If your Pattern is only one day long, the Start Date/Time's date is irrelevant.</p>
Duration (Days)	The length of the Schedule Pattern in days. This setting determines how often the Pattern repeats.
Effective Date/Time	The date and time the Schedule Pattern takes effect. LOGICTrak will not begin using this Pattern until this date and time. With this setting, you can phase in new Schedule Patterns without manually restarting the LOGICTrak Service.
Item Color Type	The color scheme to use for displaying Schedule Pattern Items. Select either "Shift Color" to use the colors of the Shifts, or "Team Color" to use the colors of the Teams.
Calendar Tick Marks (Minutes)	The vertical scale of the Calendar for editing Schedule Pattern Items. This adjusts the appearance of the Schedule

	Pattern Item editor. A tick mark will appear on the left side of the editor for every interval of this setting's length.
--	--------------------------------------------------------------------------------------------------------------------------

## Schedule Pattern Items

Each Item within a Schedule Pattern represents a single instance of a Shift and indicates whether or not it is considered "scheduled" time for KPI purposes.

### Properties

General	
Start Date/Time	The date and time this Shift instance starts, relative to the parent Schedule Pattern's Start Date/Time.
End Date/Time	The date and time this Shift instance ends, relative to the parent Schedule Pattern's Start Date/Time.
Shift Date	The date for this Shift instance's data. When a Shift spans midnight, you may want to relate the data to the first date or the second date, depending on your needs. Select "Based on Start Date" for data to use the date on which the Shift starts; select "Based on End Date" for data to use the date on which the Shift ends.
Shift	The Shift associated with this part of the Schedule. If a Shift occurs multiple times during the Schedule, you must create a Schedule Pattern Item for each instance.
Team	The Team working the specified Shift. In some cases, different Teams may work the same Shift different times of the week.
Scheduled	When checked, this setting indicates that this Pattern Item represents scheduled production time for KPI purposes. Any Availability Loss events that occur during this period should adversely impact OEE Availability. For periods of non-production, such as the weekend, you can explicitly create an unscheduled Pattern Item or simply leave a blank gap. Gaps in your Schedule imply unscheduled.

## Shifts

Shifts are reoccurring periods of time. You define your Shift Schedules using Schedule Patterns and Schedule Pattern Items.

### Properties

General	
Color	This setting determines the color of this Shift's Pattern Items in the Schedule Pattern's Item editor. In order for this setting to take effect, you must set the Schedule Pattern's Item Color Type to "Shift Color."

## Teams

Teams represent groups of operators. In some cases, different Teams work the same Shift later in the week. To accommodate this, TrakSYS allows you to specify a separate Team and Shift combination for each Schedule Pattern Item.

## Properties

<b>General</b>	
Color	This setting determines the color of this Team's Pattern Items in the Schedule Pattern's Item editor. In order for this setting to take effect, you must set the Schedule Pattern's Item Color Type to "Team Color."
<b>Users</b>	
Available Users	Lists all Users defined in the TrakSYS configuration. Users are managed in the Administration section of MODELTrak. A User may be assigned to a Team by selecting it from this drop-down and assigning it to the Assigned Users list below.
Assigned Users	Lists the Users that are assigned to this Team.

## SCRIBETrak Item Groups

SCRIBETrak Item Groups contain one or more SCRIBETrak Items. It represents a group of options that will be displayed to an end user in the SCRIBETrak application. SCRIBETrak Item Groups are also the mechanism for securing access to manual data entry forms. Users are granted access to SCRIBETrak Item Groups and thereby all the Items within.

## Properties

<b>General</b>	
Available Users	Lists all Users defined in the TrakSYS configuration. Users are managed in the Administration section of MODELTrak. A User may be assigned to the SCRIBETrak Item Group by selecting it from this drop-down and assigning it to the Assigned Users list below.
Assigned Users	Lists the Users that are assigned rights to access the SCRIBETrak Item Group. Only Users listed here can (as well as Users marked as Administrators) open the SCRIBETrak forms within the group.

## SCRIBETrak Items

SCRIBETrak Items are data entry or editor forms that may be configured to appear in a SCRIBETrak application. The properties control how the forms will be displayed and behave to the end user. The different types of SCRIBETrak items are described in the next sections.

## Manual Data Entry Items

The Manual Data Entry SCRIBETrak Item is used for entering production and event data into TrakSYS at the end of a production run, shift or date.

Each Manual Data Entry form will contain one or more tabbed sections generically referred to as Periods. Each Period represents a contiguous duration of production that shares common inputs such as Theoretical Rate, Product and Capture values. A Job Run is one example of how Periods may be used. During data entry, the end user would create a new tab (or Period) for each Job Run within the date or Shift.

Many of the Manual Data Entry form properties contain a Visible checkbox. The Visible attribute controls the visibility of the element on the end-user data entry form. The specified default value still applies to non-visible form elements.

## Properties

<b>General</b>	
Key	A unique string that identifies this instance of the form. The Key can be used to programmatically launch the form using a command line switch from the SCRIBETrak application.
System	Specifies the System that the data entry form will write data to. This System must be marked as Manual Data Entry (from the System Advanced properties).
Start Date/Time	Specifies the default Start Date/Time for the data entry form. The Start and End Date/Time fields represent the total range for which data will be entered. This is typically set to a single day or Shift.
End Date/Time	Specifies the default End Date/Time for the data entry form. The Start and End Date/Time fields represent the total range for which data will be entered. This is typically set to a single day or Shift.
Shift	Specifies the default Shift that will be recorded for all event and production data on the form.
Team	Specifies the default Team that will be recorded for all event and production data on the form.
<b>Period</b>	
Period Name	Indicates the label that is displayed in each tab that breaks up the manual data entry form. Typically each tab on the form is used to enter in data for different production runs so the term "Job" or "Run" might be applied.
Start Date/Time	Specifies if the Start Date/Time entry field is displayed on each Period tab.
End Date/Time	Specifies if the End Date/Time entry field is displayed on each Period tab.
Duration	Specifies if the Duration entry field is displayed on each Period tab.
Product	Specifies the default Product that will be recorded for all event and production data within the active Period.
Theoretical Rate (Units/Minute)	Specifies the default Theoretical Rate for production (in terms of Units/Minute) that will be recorded for the active Period.
Theoretical Rate (Seconds/Unit)	Specifies the default Theoretical Rate for production (in terms of Seconds/Unit) that will be recorded for the active Period.
Notes	Specifies the default Notes value that will be recorded for the active Period.
Notes Visible	Specifies if the Notes entry field is displayed on each Period tab.
Notes Expanded	Specifies if the Notes entry field is expanded by default on each Period tab.
<b>Capture</b>	
Capture 01	Specifies the default value for the 1 <sup>st</sup> Capture field that will be recorded for all event and production data within the

	active Period. The actual label for the Capture 01 field will be set based on the System selected.
Capture 02	See Capture 01.
Capture 03	See Capture 01.
Capture 04	See Capture 01.
Capture 05	See Capture 01.
Capture 06	See Capture 01.
Capture 07	See Capture 01.
Capture 08	See Capture 01.
Capture 09	See Capture 01.
Capture 10	See Capture 01.
<b>Events</b>	
<b>Events Section</b>	
<b>Summary Events Section</b>	The Visible checkbox controls the visibility of the entire Summary Events data entry grid.
Average Duration Visible	Specifies if the column allowing the average duration of the Event to be entered is visible. The Average Duration is multiplied with the provided count to produce the Total Duration. Typically either the Total or the Average Duration is provided during data entry and the other is calculated.
Total Duration Visible	Specifies if the column allowing the total duration of the Event to be entered is visible. The Total Duration is divided by the provided count to produce the Average Duration. Typically either the Total or the Average Duration is provided during data entry and the other is calculated.
Notes	Specifies if the Notes column is displayed allowing entry of free form text with the data entry row.
<b>Detailed Event Section</b>	The Visible checkbox controls the visibility of the entire Detail Events data entry grid.
End Date/Time Visible	Specifies if the column allowing the End Date/Time of the Event to be entered is visible. Typically either the End Date/Time or the Duration is provided during data entry and the other is calculated.
Duration Visible	Specifies if the column allowing the Duration of the Event to be entered is visible. Typically either the End Date/Time or the Duration is provided during data entry and the other is calculated.
Event Category Visible	Specifies if the column allowing the Event Category for the Event to be selected is visible.
Notes	Specifies if the Notes column is displayed allowing entry of free form text with the data entry row.
<b>Advanced</b>	
Shift Date Type	Determines how the Event and KPI Interval Date is calculated that will be recorded with all form data. The options are the date based on the Start Date/Time or the End Date/Time specified on the form.
Duration Input Mode	Controls how duration time entry is interpreted throughout the Manual Data Entry form. Microwave mode functions much like the controls on its namesake (a microwave). Entering a value of 101 would translate to one minute and one second. Minutes mode interprets the numeric input as minutes so and entry of 101 would translate to one hour and

	41 minutes.
Show Save Confirmation	When checked the end user will be prompted with a confirmation dialog when saving data from the Manual Data Entry form.
Use Counter Grid	Specifies if the Counter Grid should be used for production counter input. The Counter Grid allows for a scrolling list of production counts to be displayed when there are too many different entries to fit on the standard sized Manual Data Entry form. If the Counter Grid is not displayed, a numeric text box is rendered for each counter entry.
Search Visible	Specifies if the search box for the Counter Grid is displayed. This setting only has an effect if the Counter Grid is displayed.
Fixed Row Count	Determines how many visible rows are displayed in the Counter Grid. This setting only has an effect if the Counter Grid is displayed.

## Event List Items

The Event List SCRIBETrak Item is used for browsing recorded event data for editing. The Event List form displays events based on the filters specified. The item properties below specify the default values for the form filters.

### Properties

General	
Key	A unique string that identifies this instance of the form. The Key can be used to programmatically launch the form using a command line switch from the SCRIBETrak application.
Area	Specifies the default Area filter value for which Events will be displayed. The user can change this value when the form is displayed.
System	Specifies the default System filter value for which Events will be displayed. The user can change this filter value when the form is displayed.
Start Date/Time	Specifies the default Start Date/Time filter value for which Events will be displayed. The user can change this filter value when the form is displayed.
End Date/Time	Specifies the default End Date/Time filter value for which Events will be displayed. The user can change this filter value when the form is displayed.
Auto Search	When checked, the events that match the default filter criteria are loaded when the form is first opened and displayed.

## KPI Interval List Items

The KPI Interval List SCRIBETrak Item is used for browsing KPI Interval data for editing. The KPI Interval List form displays intervals based on the filters specified. The item properties below specify the default values for the form filters.

### Properties

<b>General</b>	
Key	A unique string that identifies this instance of the form. The Key can be used to programmatically launch the form using a command line switch from the SCRIBETrak application.
KPI Calculation	Specifies the default KPI Calculation filter value for which Events will be displayed. The user can change this filter value when the form is displayed.
Start Date/Time	Specifies the default Start Date/Time filter value for which KPI Intervals will be displayed. The user can change this filter value when the form is displayed.
End Date/Time	Specifies the default End Date/Time filter value for which KPI Intervals will be displayed. The user can change this filter value when the form is displayed.
Auto Search	When checked, the events that match the default filter criteria are loaded when the form is first opened and displayed.

## New Event Items

The New Event SCRIBETrak Item is used for entering single new event records into the database. The item properties below specify the default values for some of the form entry fields.

### Properties

<b>General</b>	
Key	A unique string that identifies this instance of the form. The Key can be used to programmatically launch the form using a command line switch from the SCRIBETrak application.
System	Specifies the default System value for the New Event form. The user can change this value when the form is displayed.
Event Definition	Specifies the default Event Definition value for the New Event form. The user can change this value when the form is displayed.
Start Date/Time	Specifies the default Start Date/Time value for the New Event form. The user can change this value when the form is displayed.
End Date/Time	Specifies the default End Date/Time value for the New Event form. The user can change this value when the form is displayed.
Date	Specifies the default Date value for the New Event form. The user can change this value when the form is displayed.

## Subscribers

Subscribers are individuals who are eligible to receive ALERTTrak notification messages. Each Subscriber may have multiple configured Devices where they can receive the messages from subscriptions. Subscribers cannot be created directly. They are a result of creating a User in the Administrator section of MODELTrak and configuring that User as a Subscriber. Users that are configured as Subscribers will show up in the Subscriber Tree View.



Subscriber properties are the same as the User properties.

## Devices

Each Subscriber must have at least one Device configured to receive notification messages. A Device might be an SMS or mail enabled cell phone, a mail enabled PC, or any other device capable of receiving email or SMS (Text Messages).

### Properties

<b>General</b>															
Device Type	Determines the message transport protocol that will be used to send notifications to this device. The message format will be based on this selection.														
Device Address	<p>If Email is selected as the Device Type, the Device Address should be a valid email address (name@domain). If SMS* (Text Message) is selected, the address should be a valid text message target. This is typically device and carrier specific. The following table lists some common address formats for the major cell phone carriers. Note that this list is not guaranteed to be accurate and may change after the publication of this document.</p> <table><tr><td>T-Mobile</td><td>phonenumber@tmomail.net</td></tr><tr><td>Virgin Mobile</td><td>phonenumber@vmobl.com</td></tr><tr><td>AT&amp;T</td><td>phonenumber@txt.att.net</td></tr><tr><td>Cingular</td><td>phonenumber@cingularme.com</td></tr><tr><td>Sprint</td><td>phonenumber@messaging.sprintpcs.com</td></tr><tr><td>Verizon</td><td>phonenumber@vtext.com</td></tr><tr><td>Nextel</td><td>phonenumber@messaging.nextel.com</td></tr></table>	T-Mobile	phonenumber@tmomail.net	Virgin Mobile	phonenumber@vmobl.com	AT&T	phonenumber@txt.att.net	Cingular	phonenumber@cingularme.com	Sprint	phonenumber@messaging.sprintpcs.com	Verizon	phonenumber@vtext.com	Nextel	phonenumber@messaging.nextel.com
T-Mobile	phonenumber@tmomail.net														
Virgin Mobile	phonenumber@vmobl.com														
AT&T	phonenumber@txt.att.net														
Cingular	phonenumber@cingularme.com														
Sprint	phonenumber@messaging.sprintpcs.com														
Verizon	phonenumber@vtext.com														
Nextel	phonenumber@messaging.nextel.com														

\* Contact your service provider for limitations on the use of SMS text messaging via email.

## Subscriptions

Subscriptions represent a request for a message to be sent to a single User on a specified Device. The Subscription contains parameters that govern the conditions under which the notification is to be sent as well as the message content.

The message that is sent for each Subscription is fixed from within TrakSYS. The standard message content is different depending on the Device type (SMS or Email). Custom text can be added to the standard message using some of the configuration properties below.

The following properties are common to all Subscriptions.

### Properties

<b>Message</b>	
Pre Message Custom Text	Specifies custom text that is appended to the beginning of the standard notification content.
Post Message Custom Text	Specifies custom text that is appended to the end of the standard notification content.
<b>Advanced</b>	

Report URL Base	Contains the base URL that is used when the standard notification message contains links to TrakSYS reports. This setting will only need to be changed when the Reporting Services instance is installed on a different server than the TrakSYS database.
From Email Address	Specifies the address that will be displayed in the From field of emails that are sent for the Subscription.

## Event Start Subscriptions

The Event Start Subscription allows notifications to be sent on the starting edge of the specified Event Definition(s). The following properties specify the Event Definitions and conditions which will trigger the notification.

### Properties

<b>General</b>	
System	Specifies the System for which notifications will be sent.
Sub-System	Specifies the Sub-System for which notifications will be sent. If All Sub-Systems is specified then notifications will be sent for all Event Definitions in all Sub-Systems for the selected System.
Event Definition	Specifies the Event Definition for which notifications will be sent. If All Event Definitions is specified then notifications will be sent for all Event Definitions in the selected System / Sub-System selected above.
Shift	Specifies a Shift filter for notifications. Only Events that take place for the specified Shift will generate notifications.
Team	Specifies a Team filter for notifications. Only Events that take place for the specified Team will generate notifications.
Product	Specifies a Product filter for notifications. Only Events that take place for the specified Product will generate notifications.
OEE Type	Specifies an OEE Type filter for notifications. Only Events that take place for the specified OEE Type will generate notifications.
Minimum Duration Seconds	Defines the minimum duration in seconds that an Event must last before the notification is sent. A value of 0 indicates that the notification should be sent as soon as the Event starts. A notification is never sent if an Event is ended before it reaches the Minimum Duration Seconds value.
Seconds Since Last Message	Defines the minimum time that must pass between a previous notification for this Subscription and the next notification that is sent. The purpose of this setting is to prevent a flood of messages being sent out when an Event Definition is triggered in rapid succession. For example, if the Seconds Since Last Message is set to 600, a notification message will be sent out a maximum of once every 10 minutes even if the Event Definition specified is triggered every 30 seconds.

## Event End Subscriptions

The Event End Subscription allows notifications to be sent on the trailing (ending) edge of the specified Event Definition(s). The following properties specify the Event Definitions and conditions which will trigger the notification.

### Properties

<b>General</b>	
System	Specifies the System for which notifications will be sent.
Sub-System	Specifies the Sub-System for which notifications will be sent. If All Sub-Systems is specified then notifications will be sent for all Event Definitions in all Sub-Systems for the selected System.
Event Definition	Specifies the Event Definition for which notifications will be sent. If All Event Definitions is specified then notifications will be sent for all Event Definitions in the selected System / Sub-System selected above.
Shift	Specifies a Shift filter for notifications. Only Events that take place for the specified Shift will generate notifications.
Team	Specifies a Team filter for notifications. Only Events that take place for the specified Team will generate notifications.
Product	Specifies a Product filter for notifications. Only Events that take place for the specified Product will generate notifications.
OEE Type	Specifies an OEE Type filter for notifications. Only Events that take place for the specified OEE Type will generate notifications.
Minimum Duration Seconds	Defines the minimum duration in seconds that an Event must last before the notification is sent. A value of 0 indicates that the notification should be sent as soon as the Event starts. A notification is never sent if an Event is ended before it reaches the Minimum Duration Seconds value.
Seconds Since Last Message	Defines the minimum time that must pass between a previous notification for this Subscription and the next notification that is sent. The purpose of this setting is to prevent a flood of messages being sent out when an Event Definition is triggered in rapid succession. For example, if the Seconds Since Last Message is set to 600, a notification message will be sent out a maximum of once every 10 minutes even if the Event Definition specified is triggered every 30 seconds.

## Conditional Event Subscriptions

The Conditional Event Subscription allows notifications to be sent when an Event Definition is triggered a specified number of time within the configured time period. It allows notifications to be sent for conditions like "the Labeler faults 3 times within 10 minutes". The condition is evaluated based on the Start Date/Time (leading edge) of the Events falling within the specified time range. The following properties specify the Event Definitions and conditions which will trigger the notification.

### Properties

<b>General</b>	
System	Specifies the System for which notifications will be sent.
Sub-System	Specifies the Sub-System for which notifications will be sent. If All Sub-Systems is specified then notifications will be sent for all Event Definitions in all Sub-Systems for the selected System.
Event Definition	Specifies the Event Definition for which notifications will be sent. If All Event Definitions is specified then notifications will be sent for all Event Definitions in the selected System / Sub-System selected above.
Shift	Specifies a Shift filter for notifications. Only Events that take place for the specified Shift will generate notifications.
Team	Specifies a Team filter for notifications. Only Events that take place for the specified Team will generate notifications.
Product	Specifies a Product filter for notifications. Only Events that take place for the specified Product will generate notifications.
OEE Type	Specifies an OEE Type filter for notifications. Only Events that take place for the specified OEE Type will generate notifications.
Minimum Duration Seconds	Defines the minimum duration in seconds that an Event must last before the notification is sent. A value of 0 indicates that the notification should be sent as soon as the Event starts. A notification is never sent if an Event is ended before it reaches the Minimum Duration Seconds value.
Seconds Since Last Message	Defines the minimum time that must pass between a previous notification for this Subscription and the next notification that is sent. The purpose of this setting is to prevent a flood of messages being sent out when an Event Definition is triggered in rapid succession. For example, if the Seconds Since Last Message is set to 600, a notification message will be sent out a maximum of once every 10 minutes even if the Event Definition specified is triggered every 30 seconds.
Number of Events	Specifies the event count that must be accumulated within the time specified by the In Last N Seconds before a notification is sent for this Subscription.
In Last N Seconds	Specifies the time frame in which the event count specified by the Number of Events must be accumulated before a notification is sent for this Subscription.

## KPI Subscriptions

The KPI Subscription allows notifications to be sent when a KPI value exceeds a specified setpoint. Multiple KPIs and setpoints can be set within the same Subscription definition to form conditions such as "notify me when the Performance falls below 60% and the Quality falls below 80%". The following properties specify the KPI Calculation and conditions which will trigger the notification.

### Properties

<b>General</b>	
KPI Calculation	Specifies the KPI Calculation for which notifications will be

	sent.
Period Type	Indicates the period of time that is monitored for KPI threshold violation. Options include Current Date, Current Shift or Current Interval.
Shift	Specifies a Shift filter for notifications. Only Shifts and Intervals for the specified Shift will generate notifications. This setting is not applicable when the Period Type is set to Current Date.
Team	Specifies a Team filter for notifications. Only Shifts and Intervals for the specified Team will generate notifications. This setting is not applicable when the Period Type is set to Current Date.
Minimum Total Seconds	Defines the minimum Total Duration Seconds for the selected Period Type before notifications can be generated. This setting is intended to prevent notifications being sent out for KPI violations too early in the selected Period Type.
Minimum Net Operation Seconds	Defines the minimum Net Operation Seconds (Total Seconds – System Not Scheduled Seconds) for the selected Period Type before notifications can be generated. This setting is intended to prevent notifications being sent out for KPI violations too early in the selected Period Type.
Seconds Since Last Message	Defines the minimum time that must pass between a previous notification for this Subscription and the next notification that is sent. The purpose of this setting is to prevent a flood of messages being sent out when a KPI setpoint is triggered in rapid succession. For example, if the Seconds Since Last Message is set to 600, a notification message will be sent out a maximum of once every 10 minutes even if the KPI setpoint is exceeded every 30 seconds.
<b>KPI Setpoints</b>	
OEE (less than)	When selected, a notification is sent when the cumulative OEE value for the Period Type drops below the specified setpoint.
Availability (less than)	When selected, a notification is sent when the cumulative Availability value for the Period Type drops below the specified setpoint.
Performance (less than)	When selected, a notification is sent when the cumulative Performance value for the Period Type drops below the specified setpoint.
Quality (less than)	When selected, a notification is sent when the cumulative Quality value for the Period Type drops below the specified setpoint.
TEEP (less than)	When selected, a notification is sent when the cumulative TEEP value for the Period Type drops below the specified setpoint.
Takt (greater than)	When selected, a notification is sent when the cumulative Seconds per (Calculation) Unit rises above the specified setpoint.
Bad Calculaton Units (greater than)	When selected, a notification is sent when the cumulative Bad Quality (Calculation) Units rise above the specified setpoint.

<b>KPI Display</b>	
Display OEE	When checked the OEE KPI is displayed in the notification message that is sent as a result of this Subscription.
Display Availability	When checked the Availability KPI is displayed in the notification message that is sent as a result of this Subscription.
Display Performance	When checked the Performance KPI is displayed in the notification message that is sent as a result of this Subscription.
Display Quality	When checked the Quality KPI is displayed in the notification message that is sent as a result of this Subscription.
Display TEEP	When checked the TEEP KPI is displayed in the notification message that is sent as a result of this Subscription.
Display Takt	When checked the Takt KPI is displayed in the notification message that is sent as a result of this Subscription.
Display Production	When checked the Production Counts are displayed in the notification message that is sent as a result of this Subscription.
Display Time	When checked the Time breakdown is displayed in the notification message that is sent as a result of this Subscription.

## Tag Subscriptions

The Tag Subscription allows notifications to be sent when a Tag becomes equal to a specified value. The following properties specify the Tag and value which will trigger the notification.

### Properties

<b>General</b>	
Tag	Specifies the Tag that is monitored for value changes.
Notify When Tag Equals	Specifies the target value for the configured Tag that will trigger a notification from this Subscription.
Seconds Since Last Message	Defines the minimum time that must pass between a previous notification for this Subscription and the next notification that is sent. The purpose of this setting is to prevent a flood of messages being sent out when a Tag value condition is met many times in rapid succession. For example, if the Seconds Since Last Message is set to 600, a notification message will be sent out a maximum of once every 10 minutes even if the tag value condition becomes True every 30 seconds.

## Users

TrakSYS Users represent individuals that may have access to WEBTrak and SCRIBETrak. They are also required for creating ALERTTrak Subscribers.

A TrakSYS User may be mapped to a Windows User/Group or be defined as an internal User where the login and password are managed within MODELTrak.

## Properties

General	
External ID	Allows the storage and lookup of a business specific user identifier, which can be used as an alternative to the users name or login to verify logins or interoperability with external applications. Company specific employee ID numbers or codes from magnetic employee badges are examples of this type of alternate identifier that helps TrakSYS™ integrate with other business systems for data and configuration exchange.
Type	A User Type categorizing this User (solely for organizational and reporting purposes).
Login Type	<p>Specifies the type of User defined.</p> <p>Windows User ties the User to an existing Windows account. The login and password for Windows Users is maintained in the Windows Active Directory.</p> <p>Windows Group ties the User to an existing Windows Group definition. The Group membership Windows Groups are maintained in the Windows Active Directory.</p> <p>TrakSYS User specifies that the user login and password information are to be maintained within MODELTrak.</p> <p>TrakSYS Group allows individual Users to be grouped for reporting and login control purposes.</p> <p>IP Address is used when access to WEBTrak is required to be controlled by node and the computers do not have Windows PC names. IP Address Users are only useable in assigning permissions in WEBTrak.</p> <p>Computer Name is used when access to WEBTrak is required to be controlled by node and the computers can be referenced by Windows PC names. Computer Name Users are only useable in assigning permissions in WEBTrak.</p> <p>Identifier is used when access to WEBTrak is required to be controlled by node and the computers do not have Windows PC names or consistent IP Addresses. The Identifier can be passed into any WEBTrak URL by adding the QueryString variable Identifier=XXX where XXX is the unique string Identifier specified below. Identifier Users are only useable in assigning permissions in WEBTrak.</p> <p>Everyone is a special User Type group which will include all users attempting to access a resource. It can be used as a simple way to assign permissions to all users attempting access. It is analogous to the EVERYONE Windows Group but also includes access when the Windows User is not known or empty.</p>

User	Specifies the Windows User name if the User is defined as a Windows User. The value specified must be unique within the User list.
Group	Specifies the Windows or TrakSYS Group name if the User is defined as a Windows or TrakSYS Group. The value specified must be unique within the User list.
Login	Specifies the username if the User is defined as a TrakSYS User. The value specified must be unique within the User list. Logins are not case sensitive when being entered into TrakSYS login screens.
IP Address	Specifies the IP Address of the computer if the User is defined as an IP Address. The value specified must be unique within the User list.
Computer Name	Specifies the network computer name if the User is defined as a Computer Name. The value specified must be unique within the User list.
Identifier	Specifies the unique string identifier if the User is defined as an Identifier. The value specified must be unique within the User list.
Password	Specifies the password if the User is defined as a TrakSYS User. Passwords are case sensitive when being entered into TrakSYS login screens.
Administrator	Indicates that the User is a TrakSYS Administrator. A TrakSYS Administrator has the highest privileges possible within all of the TrakSYS applications.
Auditor	Indicates that the User is authorized to sign off changes when the AUDITTrak digital signature dialog is displayed. Users without this option checked will not be permitted make changes to data or configuration when AUDITTrak is enabled.
Subscriber	Indicates that the User is an ALERTTrak Subscriber. When a User is marked as an ALERTTrak Subscriber, a node for that User will automatically be displayed in the Subscriber Tree under the ALERTTrak section of MODELTrak.
<b>Users</b>	
Available Users	Applies only to TrakSYS Group Login Types. Lists all Users defined in the TrakSYS configuration. A User may be assigned to TrakSYS Groups by selecting it from this drop-down and assigning it to the Assigned Users list below.
Assigned Users	Applies only to TrakSYS Group Login Types. Lists the Users that belong to TrakSYS Group.
<b>Advanced</b>	
Locale	Specifies the Users preferred locale. The locale string syntax is the based on standard the Microsoft locale list (English US = en-us).

## Script Library

The Script Library is where re-useable code can be defined and stored. Each Script entry here can contain namespaces and classes written in the C#.NET language. Publically declared objects can be referenced from within Advanced Script Tags and the Script sections of the LOGICTrak Properties page.



The Script Library is only available when the Advanced Script Tag feature is licensed. See also Script Class Templates.

### Properties

General	
Script	Holds the script content for the Script entry.

## Custom Property Schemes

Custom Property Schemes define a set of Custom Properties that may be applied to one or more TrakSYS entity types. Custom Properties can be used to define additional attributes for informational purposes and use in writing custom reports. Custom Properties can also be accessed by LOGICTrak during scans and used as capture values or in Event Definition and KPI Calculation evaluation.

Each Custom Property Scheme is rendered as a separate tab control in the property page for the TrakSYS entities that it is assigned to. The tab will contain all of the Custom Property Groups and Custom Properties that are defined beneath it.

The Custom Property Views must be re-generated whenever changes to Custom Property Schemes are made.

### Properties

General	
Key	A unique string that identifies the Custom Property Scheme. The Key is used in building unique field names in the Custom Property Views.
Tab Name	This string will be displayed in the tab in the Property Page for the entities that the scheme is applied to. If the Tab Name is left blank, the Custom Property Scheme Name property will be used for the Tab label.
Apply Scheme To	Specifies which TrakSYS entities the Custom Property Scheme will be applied to. A new Tab in the Property page for all selected entities will be displayed containing fields for the Custom Properties that are defined below the Scheme.
Type Filter	If the Scheme is assigned to entities with Type fields, a Type value can be entered in this field causing this Scheme to be applied to only those entities with a matching Type.

## Custom Property Groups

A Custom Property Group is a special grouping for Custom Properties within a Custom Property Scheme. Each Custom Property Group is rendered as a section within the tab for the parent Custom Property Scheme.

## Custom Properties

A Custom Property defines a new attribute that may be defined for a TrakSYS entity.

The Custom Property Views must be re-generated whenever changes to a Custom Property are made.

### Properties

<b>General</b>	
Key	A unique string that identifies the Custom Property. The Key is used in building unique field names in the Custom Property Views.
Data Type	Indicates that Data Type for the value specified for the Custom Property.
Default Value	Specifies the default value that will appear in the entry field for this Custom Property when a new TrakSYS entity is created.
Format	The string format code that will applied to data entered for this Custom Property through the TrakSYS entity Property form. The Format is only used when the Data Type is set to Float. A Format code of 0.00 indicates that there will always be 2 characters after the decimal when values for the Custom Property are displayed.
Enum Values	Lists the possible Custom Property Enums (name/value pairs) for the Custom Property. If there are Enum Values specified, the entry field for the Custom Property will be rendered as a dropdown combo on the TrakSYS entity Property form.

## Custom Property Enums

Custom Property Enums represent possible value selections for a Custom Property.

### Properties

<b>General</b>	
Value	Specifies the value that will be stored for the Custom Property when this choice is selected in the entry field rendered in the TrakSYS entity Property form.

## Custom Property Views

Custom Property Views are database views that are created in the TrakSYS database which bind Custom Properties defined to the TrakSYS entities assigned. These views list the Custom Properties as columns for the TrakSYS entities for ease of use in custom reports and other TrakSYS interoperability projects.

The Custom Property Views must be generated and re-generated manually whenever Custom Properties, Schemes or Groups are added, modified or deleted. The views can be re-generated by right-clicking the Custom Property Scheme root node in MODELTrak and selecting the "Create Custom Property Views" item from the popup menu.

When the "Create Custom Property Views" menu is clicked, a database view is created for each TrakSYS entity type that allows Custom Properties to be assigned. The views are called "vwCustomProperty<EntityType>" where <EntityType> is the name of the specific entity (Area, System, etc...).

## Groups

TrakSYS provides the following types of Groups for organizing configuration entities of the same type:

- Access Name Groups
- Event Code Groups
- Event Definition Groups
- Function Definition Groups
- EVENTTrak Node Groups
- KPI Count Category Groups
- Product Groups
- Sample Category Groups
- Sample CategoryJournal Category Groups
- Batch Quality Loss Category Groups
- Schedule Groups
- Shift Groups
- Tag Groups
- Team Groups
- Material Groups

## Types

TrakSYS allows you to define Types for the entities listed below, which allow you to arbitrarily relate those entities without affecting the configuration. Some of the Reports allow you to filter by Type.

- Area Types
- System Types
- Event Definition Types
- Event Category Types
- Event Code Types
- Function Definition Types
- Step Code Types
- Step Category Types
- Task Definition Types
- Batch Sub-System Parameter Definition Types
- Function Definition Parameter Definition Types
- Material Types
- Recipe Types
- Recipe Step Definition Types
- User Types

## System Templates

A System Template is a specially marked System in the configuration from which other Systems can inherit from. Systems that inherit from a System Template receive nearly all of the parent's configuration settings as well as the sub-entities (Event/Function/Task Definitions, KPI Calculations, etc...) and their settings. Properties that must be System specific (like Tags and other select settings) are unlocked at the inheriting System level for individual configuration. Changes made at the System Template are propagated to all of the associated children. System Templates and their children are not loaded by LOGICTrak and are not counted against the license limits.

## Creating System Templates

Creating a System Template is similar to creating non-template Systems. However there are special menu options for creating New Batch and Discrete System Templates (available when right-clicking on an Area node). Note that once created, it is not possible to convert a non-Template System to a System Template.

## Creating new Systems from Templates

A special menu option called **Create New System from Template** (available from when right-clicking on an Area node) is used to create a new inheriting System from an existing System Template. Once created, all of the applicable parent System Templates changes and configuration children are propagated to the new System.

## Adding children under a System Template

Once a System Template and inheriting Systems are created, any new configuration entities that are added to the parent System Template are propagated to the associated inheriting Systems. In turn, deleted entities under the System Template are removed from the inheriting Systems.

## Assigning Tags from Template Parents

A special technique exists for assisting in assigning Tags to inheriting Systems based on Tag naming conventions and Template Tags assigned to the parent System Template. A Tag Type called **Template Tag** allows for placeholder Tags to be created with the appropriate data types and assigned to the various Tag properties at the System Template level. A setting in the Advanced Tab of the System properties dialog allows the specification of a **Template Tag Prefix**. This setting stores the common prefix characters for all Tags uses at the System Template level. In turn the **Template Tag Prefix** setting at on the inheriting Systems is set with the common prefix characters for all Tags that will be assigned respectively. The Template Tags are created and assigned to the System Template entities. Real Tags are created for each child System. A special menu option called **Assign Tags from System Template** (available by right-clicking an inheriting System node) uses the Template Tag Prefix settings from the Template and inheriting Systems to match Tags and auto assign the appropriate Tags to the inheriting children and their sub-entities. The following example illustrates a simple scenario.

A System Template is created called **Packaging Template**. Its Template Tag Prefix setting is configured as **PT\_**. A Template Tag is created called **PT\_JOB** and assigned as the Job Tag in the System Template.

Two inheriting Systems are created called **Packaging Line 1** and **Packaging Line 2**. Their Template Tag Prefix settings are set to **P1\_** and **P2\_** respectively. Real OPC Tags are created called **P1\_JOB** and **P2\_JOB**.

Right clicking on the inheriting Systems and selecting the **Assign Tags from System Template** will auto find and assign the **P1\_JOB** and **P2\_JOB** Tags to the System properties. The results of the Tag assignments are displayed in a dialog and can be optionally exported to a CSV file for review.

## Advanced Script Tag API

From the code window inside Advanced (C#) Script Tags, Script Libraries, and the LOGICTrak events (Initialize, Pre-Scan and Post-Scan), a global API can be accessed allowing the use of several built-in utility functions. The following sections describe the API and its capabilities.

The API is accessed through a static class called *Global*. The *Global* class contains two static members called *Context* and *Tags*.

### Global.Context

The *Context* object contains miscellaneous utility methods as described below.

#### CreateSampleSubGroup

```
public bool CreateSampleSubGroup(  
    int sampleDefinitionID,  
    ETS.Core.SampleSubGroupItem[] samples  
)  
  
public bool CreateSampleSubGroup(  
    int sampleDefinitionID,  
    ETS.Core.SampleSubGroupData data,  
    int sampleCount,  
    ETS.Core.SampleSubGroupItem[] samples  
)
```

This method is used to programmatically add SPC data to the TrakSYS database. The second overload is used only when the Sample Definition configured as Percent mode. The return value is False if the action fails for any reason.

#### sampleDefinitionID

The ID for the Sample Definition that data will be added to.

#### data

An ETS.Core.SampleSubGroupData object. This object is loaded with additional sub-group header information for the entire entry.

#### sampleCount

The total number of samples tested for the sub-group.

#### samples

An array of ETS.Core.SampleSubGroupItem objects. Each object represents an individual measurement.

#### CreateTagHistoryRecord

```
public bool CreateTagHistoryRecord(  
    int tagHistoryDefinitionID,  
    ETS.Core.Strings.Variant valueToStore  
)
```

This method is used to programmatically add an historical Tag value record to the TrakSYS database. This is typically used when the target Tag History Definition is configured as Manual. The return value is False if the action fails for any reason.

### **tagHistoryDefinitionID**

The ID for the Tag History Definition that the record will be added for.

### **valueToStore**

The actual Tag value to store. This parameter is of type ETS.Core.Strings.Variant. It should be used as follows...

```
new ETS.Core.Strings.Variant("string value"); // string
new ETS.Core.Strings.Variant(3); // int
new ETS.Core.Strings.Variant(3.1); // double
new ETS.Core.Strings.Variant(true); // discrete
```

### **GetDataTable**

```
public System.Data.DataTable GetDataTable(
    string sql
)
```

This method returns a standard .NET DataTable object using the specified query against the TrakSYS database.

### **sql**

The SQL input query used to retrieve the DataTable.

### **GetOrCreateVariable**

```
public object GetOrCreateVariable (
    string name,
    object defaultValue
)
```

This method is used to retrieve a variable from the in-memory global variable store for the LOGICTrak engine. Value can be set to the variable store using the *SetVariable* method (see below). The value is returned as an object and must be casted to the proper type.

### **name**

The name of the variable to be retrieved.

### **defaultValue**

The default value for the variable to be retrieved. If the requested variable does not exist, it is created in the store and the *defaultValue* is returned.

### **GetNonPrerequisiteTag**

```
public ETS.Core.Scripting.ISourceCodeTag GetNonPrerequisiteTag (
    string name
)
```

From script inside an Advanced Script Tag, the Tags collection (below) contains only Tags that are explicitly referenced. This method allows access to Tag values that are not explicitly referenced by the Script Tag. It should be noted though that Tags retrieved using this method are not guaranteed to have been evaluated during the current LOGICTrak scan. This method returns a Tag object.

**name**

The name of the Tag which will be retrieved.

**GetSampleSubGroupDataObject**

```
public SampleSubGroupData GetSampleSubGroupDataObject (
    int sampleDefinitionID
)
```

This method creates and returns a *SampleSubGroupData* object for the specified Sample Definition. The object contains current values for properties such as Shift, Capture, Product and limits.

**sampleDefinitionID**

The ID of the Sample Definition for which to load the return object.

**GetTagIDFromName**

```
public int GetTagIDFromName (
    string name
)
```

This method looks up a Tag's ID based on its name. The return value is the Tag ID.

**name**

The name of the Tag for which the ID will be retrieved.

**GetTagNameFromID**

```
public string GetTagNameFromID (
    int tagID
)
```

This method looks up a Tag's name based on its ID. The return value is the Tag name.

**tagID**

The ID of the Tag for which the name will be retrieved.

**GetVariable**

```
public object GetVariable (
    string name
)
```

This method is used to retrieve a variable from the in-memory global variable store for the LOGICTrak engine. Value can be set to the variable store using the *SetVariable* method (see below). The value is returned as an object and must be casted to the proper type. This method returns null if the requested variable does not exist.

**name**

The name of the variable to be retrieved.

## HasVariable

```
public bool HasVariable (
    string name
)
```

This method tests the in-memory global variable store for the LOGICTrak engine for the existence of the specified variable. The return value is True if the variable exists.

### name

The name of the variable to test for.

## Divide

```
public double HasVariable (
    double numerator,
    double divisor,
    double ifNan
)
```

This method divides the specified *divisor* by the *numerator* and returns the result. If the result is not a number (NaN) in the case of a divide by zero or other condition, the method returns the value specified by the ifNan parameter.

### numerator

The numerator value in the division operation.

### divisor

The divisor value in the division operation.

### ifNan

The value returned if the result of the division is NaN.

## QueueTagForAssignment

```
public bool QueueTagForAssignment (
    int tagID,
    ETS.Core.Enums.TagAssignOperation operation,
    ETS.Core.Strings.Variant value
)
public bool QueueTagForAssignment (
    string tagName,
    ETS.Core.Enums.TagAssignOperation operation,
    ETS.Core.Strings.Variant value
)
```

This method allows for the programmatic setting of a Tag's value. When this method is called from within a LOGICTrak scan (rather than between), the Tag value assignment will not take place until after the current scan ends. The method returns False if the assignment queue fails.

### tagID

The ID for the Tag to be assigned.

### tagName

The name for the Tag to be assigned.



### **operation**

This enum determines how the assigned value is applied to the existing Tag value. Options are Add, Concatenate, Divide, Equals, Modulus, Multiply and Subtract.

### **value**

The value to be assigned. This parameter is of type ETS.Core.Strings.Variant. It should be used as follows...

```
new ETS.Core.Strings.Variant("string value"); // string  
new ETS.Core.Strings.Variant(3)); // int  
new ETS.Core.Strings.Variant(3.1)); // double  
new ETS.Core.Strings.Variant(true)); // discrete
```

### **QueueTagForEvaluationNextScan**

```
public void QueueTagForEvaluationNextScan (  
    int tagID  
)
```

This method forces the specified Tag to be evaluated in the next LOGICTrak scan, regardless of whether its value changes or not.

### **tagID**

The ID for the Tag to be queued.

### **QueueTagForEvaluationThisScan**

```
public void QueueTagForEvaluationThisScan (  
    int tagID  
)
```

This method forces the specified Tag to be evaluated later in the current LOGICTrak scan, regardless of whether its value changes or not.

### **tagID**

The ID for the Tag to be queued.

### **QueueTagNameEqualsForAssignment**

```
public bool QueueTagNameEqualsForAssignment (  
    string tagName,  
    ETS.Core.Strings.Variant value  
)  
public bool QueueTagNameEqualsForAssignment (  
    string tagName,  
    int value  
)  
public bool QueueTagNameEqualsForAssignment (  
    string tagName,  
    double value  
)  
public bool QueueTagNameEqualsForAssignment (  
    string tagName,  
    string value  
)  
public bool QueueTagNameEqualsForAssignment (  
    string tagName,  
    bool value
```

```
)
```

This method allows for the programmatic setting of a Tag's value. When this method is called from within a LOGICTrak scan (rather than between), the Tag value assignment will not take place until after the current scan ends. The method returns False if the assignment queue fails.

**tagName**

The name for the Tag to be queued.

**value**

The value to be assigned. The following examples apply when the overload using the type ETS.Core.Strings.Variant is used.

```
new ETS.Core.Strings.Variant("string value"); // string
new ETS.Core.Strings.Variant(3); // int
new ETS.Core.Strings.Variant(3.1); // double
new ETS.Core.Strings.Variant(true); // discrete
```

**SaveVariable**

```
public void SaveVariable (
    string name,
    object newValue
)
```

This method sets the a value to an in-memory global variable store for the LOGICTrak engine. The value can be retrieved using the *GetVariable* method (above).

**name**

The name of the variable to be set.

**newValue**

The value to be set to the specified variable.

**Now (Property)**

```
public DateTime Now { get; }
```

This property returns the current date and time used internally in the LOGICTrak engine. This timestamp is frozen in place for the within the execution of each scan.

**ScanMode (Property)**

```
public ETS.Core.Enums.LogicManagersScanType ScanMode { get; }
```

This property returns an enum that indicates the type of LOGICTrak scan that is in process. The possible values are ScanNone, ScanFirst, ScanNormal, ScanCheckDB and ScanLast.

**Global.Tags**

The *Tags* object is a collection that contains the individual Tag objects that are loaded into LOGICTrak. A specific Tag in the collection can be referenced using the

following syntax ... *Global.Tags["tagname"]*. Each Tag object in the collection is of type *ISourceCodeTag* and contains the following properties.

### **ID**

```
public int ID { get; }
```

Returns the ID of the Tag.

### **Name**

```
public string Name { get; }
```

Returns the name of the Tag.

### **DataType**

```
public int ETS.Core.Enums.DataType { get; }
```

Returns the TrakSYS Tag data type. Possible values are String, Integer, Float or Discrete.

### **TagType**

```
public int ETS.Core.Enums.TagType { get; }
```

Returns the TrakSYS Tag type for the Tag.

### **Quality**

```
public int ID { get; }
```

Returns the current Quality for the Tag. This property is typically used for OPC or other externally acquired Tags. A value of 192 is typically good where any other value is bad, however the actual Quality values are source dependent.

### **ValueString**

```
public string ValueString { get; }
```

Returns the current value of the Tag casted to a string data type.

### **ValueInteger**

```
public int ValueInteger { get; }
```

Returns the current value of the Tag casted to an int data type.

### **ValueDouble**

```
public double ValueDouble { get; }
```

Returns the current value of the Tag casted to a double data type.

### **ValueBoolean**

```
public bool ValueBoolean { get; }
```

Returns the current value of the Tag casted to a bool data type.

## **Script Class Templates**

Script Classes can be defined to extend the functionality of several key configuration entities. By defining a Script Class that inherits from a specific base class, code can be executed at various key stages in the LOGICTrak scan execution. Once a specialized inheriting Script Class is defined, it can be associated with specific entities by configuring the entity's Script Class Name property in the Advanced tab of its properties dialog. The empty class structure for specialized inheriting Script Classes can be created easily by using the second level menu fly-out when creating new Script Classes. The following special menu options create the extension and methods.

### **LOGICTrak Script Class**

This class allows code to be executed on LOGICTrak startup, in the pre-scan and in the post scan. This class replaced the built in scripting sections in the LOGICTrak property dialog (from version 6.X). During an upgrade from 6.X, the code in the LOGICTrak scripting sections is automatically moved to a LOGICTrak Script Class.

### **System Script Class**

This class allows code to be executed for a specific System during the LOGICTrak startup, in the pre-scan and in the post scan. In addition, several System specific code events can be overridden and used. Note that in all of the Methods that begin with PostScan..., the Method is actually being triggered in the post scan timeframe (rather than in mid-scan as the events occur). A context argument is passed into to each Method with common API functions as well as Method specific data.

### **Event Definition Script Class**

This class allows code to be executed for a specific Event Definition during the LOGICTrak startup, in the pre-scan and in the post scan. In addition, several Event Definition specific code events can be overridden and used. Note that in all of the Methods that begin with PostScan..., the Method is actually being triggered in the post scan timeframe (rather than in mid-scan as the events occur). A context argument is passed into to each Method with common API functions as well as Method specific data.

### **Function Definition Script Class**

This class allows code to be executed for a specific Function Definition during the LOGICTrak startup, in the pre-scan and in the post scan. In addition, several Function Definition specific code events can be overridden and used. Note that in all of the Methods that begin with PostScan..., the Method is actually being triggered in the post scan timeframe (rather than in mid-scan as the events occur). A context argument is passed into to each Method with common API functions as well as Method specific data.

### **Task Definition Script Class**

This class allows code to be executed for a specific Task Definition during the LOGICTrak startup, in the pre-scan and in the post scan. In addition, several Task

Definition specific code events can be overridden and used. Note that in all of the Methods that begin with PostScan..., the Method is actually being triggered in the post scan timeframe (rather than in mid-scan as the events occur). A context argument is passed into to each Method with common API functions as well as Method specific data.

## **KPI Calculation Script Class**

This class allows code to be executed for a specific KPI Calculation during the LOGICTrak startup, in the pre-scan and in the post scan. In addition, several KPI Calculation specific code events can be overridden and used. Note that in all of the Methods that begin with PostScan..., the Method is actually being triggered in the post scan timeframe (rather than in mid-scan as the events occur). A context argument is passed into to each Method with common API functions as well as Method specific data.

## **LOGICTrak**

The LOGICTrak is a Windows service that runs in the background collecting I/O, recording Events, tracking KPI, and sending event notifications to EVENTTrak nodes. Since LOGICTrak typically runs on a server without any users logged in, the service itself does not provide a user interface; you must use the LOGICTrak Client application to view the current status of the LOGICTrak Service as well as the Tags being monitored. The sections below describe the LOGICTrak Service and Client in more detail.

## **LOGICTrak Service**

The LOGICTrak Service runs in the background, collecting I/O and recording data to the database. It has no user interface because it typically runs on a server where no user is logged in. You must use the LOGICTrak Client application to view its current status.

## **Starting and Stopping the Service**

You can start and stop the LOGICTrak Service directly from the Windows Services MMC snap-in located in Administrative Tools, or you can use LOGICTrak Client. When using the Services snap-in, search for the service named "LOGICTrak" in the service list.

A common use of the start and stop commands are to apply changes to the configuration. The LOGICTrak Service loads its configuration from the database upon startup. It does not reload the configuration while it is running, so you must restart it in order for changes to take effect.

## **Configuring the Service**

The Windows Services MMC snap-in, located in Administrative Tools, provides additional configuration settings for the LOGICTrak Service beyond those available in MODELTrak. To modify these settings, open the properties page for the "LOGICTrak" service in the Services list.

If you would like LOGICTrak to automatically begin running when Windows starts, select the General tab and set the Startup type to Automatic. Otherwise, if the

Startup type is Manual, a user must log in and manually start the service via the Services snap-in or LOGICTrak Client.

To change the security credentials LOGICTrak uses, select the "Log On" tab and specify the Windows user account of your choice. LOGICTrak does not use the service account to connect to the database, but it does use the account for local file access and DCOM. For more information on troubleshooting DCOM security, refer to the OPC Troubleshooting Guide's section on COM Security.

## LOGICTrak Client

LOGICTrak Client is a standalone application which allows you to view the current status of the LOGICTrak Service, which may be on the same computer or on a remote server.

The top of the LOGICTrak Client's tree view displays all configured LOGICTrak Instances, allowing you to view each service's state and statistics. By right-clicking on the corresponding LOGICTrak node, you can start, stop, or restart an instance of the LOGICTrak Service.

The list view displays the following information about the selected LOGICTrak Instance:

Statistics	
Status	Shows the current state of the LOGICTrak Service. Possible values are Started, Stopped or Starting.
Scan Date/Time	The date/time of the last LOGICTrak scan. This timestamp is based on the clock from the server the database is running on.
Scan Duration (Milliseconds)	The duration (in milliseconds) of the last scan LOGICTrak executed. This does not include the idle time between scans.
Scan Interval (Milliseconds)	The duration (in milliseconds) of the last interval between LOGICTrak scans. There is no processing of any logic during this time.
Scan Tags	The number of Tags that changed value in the last scan.
Scan Event Definitions	The number of Event Definition conditions (events) that started or ended in the last scan.
Simulation Mode	Specifies if the LOGICTrak is configured and running in Simulation Mode. In Simulation Mode, none of the OPC or Database Tags are connected.
Load Tags Only	Specifies if the LOGICTrak is configured to load only Tags. When Load Tags Only is on, only Tags are loaded and connected. Systems and other business logic entities are not loaded and executed. This mode is typically used to

	test connection to Tags.
Load All Tags	Specifies that if the LOGICTrak is configured to load all Tags or only the Tags that are being utilized by other configuration entities.

The tree view also displays all configured Tags with the same hierarchical view as in MODELTrak. Each Tag includes the following information:

Name	The name of the Tag.
Data Type	The Tag's configured Data Type (Discrete, Integer, Float, String).
Type	The Tag's configured Type (OPC, Compare, Virtual, etc...).
Value	The Tag's current effective value.
Quality	An integer representing the Tag's quality. Good quality is typically 192.
Updated	The date and time the Tag's value last changed.
Item Name	The OPC Tag's address of an item on the OPC server.

LOGICTrak Client will only display correct values for Tags with the Persist Value to Database option enabled. Also, the LOGICTrak Service must be running and the Tags must be referenced by the loaded configuration if the Load All Tags or Load Tags Only options are disabled; otherwise LOGICTrak Client will display "Not Loaded."

The Reload Tree menu option, accessible from File menu, allows you to manually refresh the Tag Group tree. This is useful when new Tag Groups are added to the configuration.

The Reset Tag Quality menu option, accessible from the File menu, sets the Quality values of all Tags to 192 (Good). This is useful when Tag Qualities are incorrect due to switching back and forth from Simulation and non-Simulation modes.

The Refresh Rates settings, accessible from the Settings menu, allow you to adjust how frequently LOGICTrak Client scans the database for updates. This dialog includes the following settings:

<b>Refresh Rates</b>	
LOGICTrak Refresh (Milliseconds)	The frequency (in milliseconds) at which LOGICTrak Client scans the database for updated status information on the LOGICTrak Service.
LOGICTrak Message Refresh (Milliseconds)	The frequency (in milliseconds) at which the messages in the LOGICTrak Messages node are refreshed.
Tag Refresh (Milliseconds)	The frequency (in milliseconds) at which LOGICTrak Client scans the database for updated Tag values and qualities.
EVENTTrak Refresh (Millisecond)	The frequency (in milliseconds) at which the messages in the EVENTTrak View nodes are refreshed.

Another menu option accessible from the Settings menu is Show Not Loaded Tags. This allows showing and hiding of the Tags that are not loaded by the LOGICTrak.

For convenience, LOGICTrak Client also provides shortcuts to the Database Manager and License Manager, accessible via the Tools menu item at the top of the application. With these options you can quickly adjust the active database settings or verify what license features you have installed.

The LOGICTrak Messages node in the tree view lists the LOGICTrak specific error log entries. This node is useful when troubleshooting a failed LOGICTrak start or restart.

The Tag View nodes in the tree view display the real-time values of the Tags loaded and monitored by the LOGICTrak service(s).

The EVENTTrak View nodes in the tree view displays the EVENTTrak messages that are present for each configured EVENTTrak client. These nodes are useful when troubleshooting EVENTTrak communication issues.

## EVENTTrak

EVENTTrak is a set of ActiveX controls which run within an ActiveX container application. Supported container environments include Visual Basic, C#, InTouch, and others. EVENTTrak alerts operators to recent and ongoing production events such as stoppages, rejects, and slowdowns, and allows them to acknowledge, categorize, and annotate these events according to the configuration.

Starting in version 7, the EVENTTrak user interaction functionality has been extended to WEBTrak with the introduction of several new interactive Web Parts called EVENTTrak Parts. These new Web Parts allow complex interactive user interfaces to be built and deployed using the TrakSYS dashboard capabilities. Accessing interactive EVENTTrak functionality from WEBTrak requires that the client PCs be named nodes listed in the EVENTTrak configuration section of MODELTrak.

The following sections are programming reference for the EVENTTrak ActiveX controls, describing every property, method, and event of each control.

## Common Properties

This is a list of all the common properties that all the controls share. Most of the controls are used to help define how the control communicates with the hosting SQL Server database.

### Properties

#### DbServer

```
public string DbServer { get; set; }
```

Gets or sets the location of the Database Server. This is the computer name (or IP Address) of the computer hosting the SQL Server database.

#### DbName

```
public string DbName { get; set; }
```



Gets or sets the name of the TrakSYS database.

### **DbUsesIntegratedSecurityType**

```
public bool DbUsesIntegratedSecurityType { get; set; }
```

Gets or sets whether the database connection uses the current Windows account credentials for authentication (if True) or whether it uses the DbLogin and DbPassword properties (if False).

### **DbLogin**

```
public string DbLogin { get; set; }
```

Gets or sets the SQL Server database login. This value only applies when the DbUsesIntegratedSecurityType property is True.

### **DbPassword**

```
public string DbPassword { get; set; }
```

Gets or sets the password for the database login to connect to the TrakSYS database. This value only applies when the DbUsesIntegratedSecurityType property is True.

### **DbConnectTimeout**

```
public int ConnectTimeout { get; set; }
```

Gets or sets the length of time (in seconds) EVENTTrak will wait for a connection to the SQL Server database before terminating the attempt.

### **DbCommandTimeout**

```
public int CommandTimeout { get; set; }
```

Gets or sets the length of time (in seconds) EVENTTrak will wait for a SQL command to execute on the SQL Server database before terminating the attempt.

### **DbConnectTemplate**

```
public string Template { get; set; }
```

Gets or sets the Connection String which establishes the connection to the TrakSYS database. This uses the various properties listed above to generate the Connection String to connect to the SQL Server database. The default Template value is:

```
Provider=SQLOLEDB.1;Password={databasePassword};User  
ID={databaseLogin};Persist Security  
Info=True;Trusted_Connection={integratedSecurity};Initial  
Catalog={databaseName};Data Source={databaseServer};Connect  
Timeout={connectTimeout}
```

EVENTTrak replaces the various tokens with the corresponding property values. For example, the DbLogin property value will replace the {databaseLogin} placeholder.

## HostType

```
public int HostType { get; set; }
```

Gets or sets how the control will be identified. The following are the only valid values:

- 1 – Computer Name
- 2 – IP Address

An EVENTTrak corresponds to a specific computer; the HostType property determines whether to identify the computer by IP Address or by Computer Name.

## Host

```
public string Host { get; }
```

Gets the name or IP address of the current host, depending on the HostType setting. When running the control in a Remote Desktop or Terminal Services session, this property will return the Host name of the Client computer rather than the Server computer.

## IsLicensed

```
public bool IsLicensed { get; }
```

Returns whether or not the current control is correctly licensed.

## Version

```
public string Version { get; }
```

Returns the current internal version of the control in the format "x.x.x". This value may be useful in troubleshooting compatibility issues.

## DbConfig Class

The DbConfig class is a helper class for assigning database properties to EVENTTrak controls. Rather than individually setting every database connection property for each control, you can simply create an instance of the DbConfig class, set all of the database properties once, and then reuse the DbConfig object to configure each control using the SetPropertiesFromDbConfig method.

## Properties

### DbServer

```
public string DbServer { get; set; }
```

Gets or sets the location of the Database Server. This is the computer name (or IP Address) of the computer hosting the SQL Server database.

## DbName

```
public string DbName { get; set; }
```

Gets or sets the name of the TrakSYS database.

## DbUsesIntegratedSecurityType

```
public bool DbUsesIntegratedSecurityType { get; set; }
```

Gets or sets whether the database connection uses the current Windows account credentials for authentication (if True) or whether it uses the DbLogin and DbPassword properties (if False).

## DbLogin

```
public string DbLogin { get; set; }
```

Gets or sets the SQL Server database login. This value only applies when the DbUsesIntegratedSecurityType property is True.

## DbPassword

```
public string DbPassword { get; set; }
```

Gets or sets the password for the database login to connect to the TrakSYS database. This value only applies when the DbUsesIntegratedSecurityType property is True.

## DbConnectTimeout

```
public int ConnectTimeout { get; set; }
```

Gets or sets the length of time (in seconds) EVENTTrak will wait for a connection to the SQL Server database before terminating the attempt.

## DbCommandTimeout

```
public int CommandTimeout { get; set; }
```

Gets or sets the length of time (in seconds) EVENTTrak will wait for a SQL command to execute on the SQL Server database before terminating the attempt.

## DbConnectTemplate

```
public string Template { get; set; }
```

Gets or sets the Connection String which establishes the connection to the TrakSYS database. This uses the various properties listed above to generate the Connection String to connect to the SQL Server database. The default template value is:

```
Provider=SQLOLEDB.1;Password={databasePassword};User  
ID={databaseLogin};Persist Security  
Info=True;Trusted_Connection={integratedSecurity};Initial
```

```
Catalog={databaseName};Data Source={databaseServer};Connect
Timeout={connectTimeout}
```

EVENTTrak replaces the various tokens with the corresponding property values. For example, the DbLogin property value will replace the {databaseLogin} placeholder.

## HostType

```
public int HostType { get; set; }
```

Gets or sets how the control will be identified. The following are the only valid values:

- 1 – Computer Name
- 2 – IP Address

An EVENTTrak corresponds to a specific computer; the HostType property determines whether to identify the computer by IP Address or by Computer Name.

## Event Category Picker Control

The Event Category Picker control allows users to select Event Categories, giving them easy access to potentially complex hierarchies of reasons and sub-reasons.

## Properties

### EventCategoryCount

```
public int EventCategoryCount { get; }
```

Returns the total number of Event Categories loaded in the control.

### Font

```
public Font Font { get; set; }
```

Gets or sets the font EVENTTrak uses to display text on the control. This is a standard object which provides access to every available font property. If the ActiveX container does not support objects, you can use the other Font properties to adjust the control's font.

### FontBold

```
public bool FontBold { get; set; }
```

Gets or sets whether or not the control's text will appear bold.

### FontCharset

```
public int FontCharset { get; set; }
```

Gets or sets the character set the control uses to display text. This property primarily helps support various language settings.

## FontName

```
public string FontName { get; set; }
```

Gets or sets the name of the font the control uses to display text.

## FontSize

```
public float FontSize { get; set; }
```

Gets or sets the font size the control uses to display text.

## ImageType

```
public int ImageType { get; set; }
```

Gets or sets the type of icon to display. The following are the only valid values:

- 0 – None
- 1 – Small Image
- 2 – Large Image

If the value is 0, there will be no image. If the value is 1, the control will use a 16 x 16 pixel image. If the value is 2, the control will use a 32 x 32 pixel image. There are separate images for Event Categories that are selectable and Event Category Groups or Event Categories that are not selectable.

## NoCategoryText

```
public string NoCategoryText { get; set; }
```

Gets or sets the display text for the TreeView node representing No Category Selected if the ShowNoCategory property is True.

## SelectedEventCategory\*ID

```
public int SelectedEventCategory01ID { get; }
```

Returns the appropriate EventCategoryID if one is selected. If the user has not selected the level of the Event Category, or if no Event Categories are selected, then this will return a value of -1. There are 10 versions of this property, representing the 10 available Event Category columns ranging from 01 to 10. There will only be values for the licensed Event Category fields.

## SelectedEventCategory\*Name

```
public string SelectedEventCategory01Name { get; }
```

Returns the appropriate EventCategoryName if one is selected. There are 10 versions of this property, representing the 10 available Event Category columns ranging from 01 to 10. There will only be values for the licensed Event Category fields.

## SelectedEventCategory\*Description

```
public string SelectedEventCategory01Description { get; }
```

Returns the appropriate EventCategoryDescription if it is selected. There are 10 versions of this property, each representing the 10 available EventCategory columns, ranging from 01 to 10. Only the licensed EventCategory fields will be set.

### **SelectedEventCategoryDescription**

```
public string SelectedEventCategoryDescription { get; }
```

Returns the SelectedEventCategoryDescription of the SelectedEventCategory.

### **SelectedEventCategoryEventCode**

```
public string SelectedEventCategoryEventCode { get; }
```

Returns the Code of the Event Code for the selected Event Category.

### **SelectedEventCategoryEventCodeID**

```
public int SelectedEventCategoryEventCodeID { get; }
```

Returns the ID of the Event Code for the selected Event Category.

### **SelectedEventCategoryEventCodeName**

```
public string SelectedEventCategoryEventCodeName { get; }
```

Returns the Name of the Event Code for the selected Event Category.

### **SelectedEventCategoryGroupID**

```
public int SelectedEventCategoryGroupID { get; }
```

Returns the ID of the selected Event Category Group. If an EventCategoryGroup is selected it will return its ID; otherwise, if an Event Category is selected, it will return the ID of the parent Event Category Group.

### **SelectedEventCategoryGroupName**

```
public string SelectedEventCategoryGroupName { get; }
```

Returns the Name of the selected Event Category Group. If an Event Category Group is selected, it will return its Name; if an Event Category is selected, it will return the parent Event Category Group's Name.

### **SelectedEventCategoryID**

```
public int SelectedEventCategoryID { get; set; }
```

Gets or sets the ID of the selected Event Category.

### **SelectedEventCategoryLevel**

```
public int SelectedEventCategoryLevel { get; }
```

Returns the Level of the selected Event Category. This corresponds to a level in the Event Category Scheme.

### **SelectedEventCategoryName**

```
public string SelectedEventCategoryName { get; }
```

Returns the Name of the selected Event Category.

### **SelectedEventCategoryOeeType**

```
public int SelectedEventCategoryOeeType { get; }
```

Returns the OeeType of the selected Event Category.

### **SelectedEventCategoryOeeTypeName**

```
public string SelectedEventCategoryOeeTypeName { get; }
```

Returns the OeeType Name of the selected Event Category.

### **SelectedEventCategorySelectable**

```
public bool SelectedEventCategorySelectable { get; }
```

Returns whether or not the selected Event Category is Selectable. This returns False if an Event Category Group is currently selected.

### **ShowGroups**

```
public bool ShowGroups { get; set; }
```

Gets or sets whether or not to show the Event Category Groups in the TreeView.

### **ShowNoCategory**

```
public bool ShowNoCategory { get; set; }
```

Gets or sets whether or not to show an entry in the TreeView representing No Event Categories Selected. If True, the node's text comes from the NoCategoryText property.

## **Methods**

### **CollapseAll**

```
public void CollapseAll()
```

This method will collapse all of the TreeView nodes, leaving nothing expanded.

## **CollapseEventCategory**

```
public void CollapseEventCategory([int EventCategoryID])
```

This method will collapse the TreeView node for the corresponding Event Category.

### **EventCategoryID**

This is the ID of the Event Category to collapse.

## **CollapseEventCategoryGroup**

```
public void CollapseEventCategoryGroup([int EventCategoryGroupID])
```

This method will collapse the TreeView node for the corresponding Event Category Group.

### **EventCategoryGroupID**

This is the ID of the Event Category Group to collapse.

## **ExpandAll**

```
public void ExpandAll()
```

This method will expand all the TreeView nodes, leaving nothing collapsed.

## **ExpandEventCategory**

```
public void ExpandEventCategory([int EventCategoryID])
```

This method will expand the TreeView node for the corresponding Event Category.

### **EventCategoryID**

This is the ID of the Event Category to expand.

## **ExpandEventCategoryGroup**

```
public void ExpandEventCategoryGroup([int EventCategoryGroupID])
```

This method will expand the TreeView node for the corresponding Event Category Group.

### **EventCategoryGroupID**

This is the ID of the EventCategoryGroup to expand.

## **GetEventCategoryIDByIndex**

```
public int GetEventCategoryIDByIndex([int Index])
```

This method will return the ID of an Event Category according to the specified index. If you omit the Index parameter, this method will return a value of -1.

### **Index**

This is the 0-based index of the appropriate item in the control.



### **GetEventCategoryDescriptionByIndex**

```
public string GetEventCategoryDescriptionByIndex([int Index])
```

This method will return the Description of the EventCategory based on the passed in parameter. If an Index is not passed in, an empty string will be returned.

#### **Index**

This is the 0-based index of the appropriate item in the control.

### **GetEventCategoryEventCodeByIndex**

```
public string GetEventCategoryEventCodeByIndex([int Index])
```

This method will return the Code of the EventCode of the EventCategory based on the passed in parameter. If an Index is not passed in, an empty string will be returned.

#### **Index**

This is the 0-based index of the appropriate item in the control.

### **GetEventCategoryEventCodeIDByIndex**

```
public int GetEventCategoryEventCodeIDByIndex([int Index])
```

This method will return the ID of the EventCode of the EventCategory based on the passed in parameter. If an Index is not passed in, a value of -1 will be returned.

#### **Index**

This is the 0-based index of the appropriate item in the control.

### **GetEventCategoryEventCodeNameByIndex**

```
public string GetEventCategoryDescriptionByIndex([int Index])
```

This method will return the Name of the EventCode of the EventCategory based on the passed in parameter. If an Index is not passed in, an empty string will be returned.

#### **Index**

This is the 0-based index of the appropriate item in the control.

### **GetEventCategoryLevelByIndex**

```
public int GetEventCategoryLevelByIndex([int Index])
```

This method will return the Level of the EventCategory based on the passed in parameter. If an Index is not passed in, a value of -1 will be returned.

#### **Index**

This is the 0-based index of the appropriate item in the control.

### **GetEventCategoryNameByIndex**

```
public string GetEventCategoryNameByIndex([int Index])
```

This method will return the Name of the EventCategory based on the passed in parameter. If an Index is not passed in, an empty string will be returned.

**Index**

This is the 0-based index of the appropriate item in the control.

**GetEventCategoryOeeTypeByIndex**

```
public int GetEventCategoryOeeTypeByIndex([int Index])
```

This method will return the OeeType of the EventCategory based on the passed in parameter. If an Index is not passed in, a value of -1 will be returned.

**Index**

This is the 0-based index of the appropriate item in the control.

**GetEventCategoryOeeTypeNameByIndex**

```
public string GetEventCategoryOeeTypeNameByIndex([int Index])
```

This method will return the OeeTypeName of the EventCategory based on the passed in parameter. If an Index is not passed in, an empty string will be returned.

**Index**

This is the 0-based index of the appropriate item in the control.

**GetEventCategoryParentEventCategoryIDByIndex**

```
public int GetEventCategoryParentEventCategoryIDByIndex([int Index])
```

This method will return the ParentEventCategoryID of the EventCategory based on the passed in parameter. If an Index is not passed in, a value of -1 will be returned.

**Index**

This is the 0-based index of the appropriate item in the control.

**GetEventCategoryParentGroupIDByIndex**

```
public int GetEventCategoryParentGroupIDByIndex([int Index])
```

This method will return the ID of the EventCategoryGroup of the EventCategory based on the passed in parameter. If an Index is not passed in, a value of -1 will be returned.

**Index**

This is the 0-based index of the appropriate item in the control.

**GetEventCategoryParentGroupNameByIndex**

```
public string GetEventCategoryOeeTypeByIndex([int Index])
```

This method will return the Name of the EventCategoryGroup of the EventCategory based on the passed in parameter. If an Index is not passed in, an empty string will be returned.

#### **Index**

This is the 0-based index of the appropriate item in the control.

#### **GetEventCategorySelectableByIndex**

```
public bool GetEventCategorySelectableByIndex([int Index])
```

This method will return the Selectable value of the EventCategory based on the passed in parameter. If an Index is not passed in, a value of false will be returned.

#### **Index**

This is the 0-based index of the appropriate item in the control.

#### **LoadByEventDefinitionID**

```
public bool LoadByEventDefinitionID(  
    [int EventDefinitionID],  
    [int SelectedEventCategoryID]  
)
```

This method will load all of the Event Categories configured for the specified Event Definition. This will load all of the appropriate Event Categories based on the current configuration in the database.

#### **EventDefinitionID**

This is the ID of the Event Definition to load.

#### **SelectedEventCategoryID**

This allows you to programmatically pre-select the Event Category by ID during the load process, which will also ensure that the selected Event Category is visible once the load is complete.

#### **MakeEventCategoryVisible**

```
public void MakeEventCategoryVisible([int EventCategoryID])
```

This method will expand all of the necessary TreeView nodes to make the specified Event Category visible in the TreeView.

#### **EventCategoryID**

This is the ID of the Event Category to be visible.

#### **SetPropertiesFromDbConfig**

```
public bool SetPropertiesFromDbConfig(DbConfig ConfigSettings)
```

This method will configure all of the control's database connection properties in one step via a reusable DbConfig object.

#### **ConfigSettings**

The DbConfig object to use to configure the database connection properties for this control.

## Events

### EventCategoryClick

```
public event EventCategoryClick(int EventCategoryID)
```

This event fires when the user clicks an Event Category in the control. This occurs regardless of whether the click is a left or right mouse click, but only if the user clicks an Event Category.

#### EventCategoryID

This is the ID of the Event Category clicked by the user.

### EventCategoryDbClick

```
public event EventCategoryDbClick(int EventCategoryID)
```

This event fires when the user double-clicks an Event Category in the control. This occurs regardless of whether the double-click is a left or right mouse double-click, but only if the user double-clicks an Event Category.

#### EventCategoryID

This is the ID of the Event Category double-clicked by the user.

### EventCategoryGroupClick

```
public event EventCategoryGroupClick(int EventCategoryGroupID)
```

This event fires when the user clicks an Event Category Group in the control. This occurs regardless of whether the click is a left or right mouse click, but only if the user clicks an Event Category Group.

#### EventCategoryGroupID

This is the ID of the Event Category Group clicked by the user.

### EventCategoryGroupDbClick

```
public event EventCategoryGroupDbClick(int EventCategoryGroupID)
```

This event fires when the user double-clicks an Event Category Group in the control. This occurs regardless of whether the double-click is a left or right mouse click, but only if the user double-clicks an Event Category Group.

#### EventCategoryGroupID

This is the ID of the Event Category Group double-clicked by the user.

### EventCategoryGroupSelected

```
public event EventCategoryGroupSelected(int EventCategoryGroupID)
```

This event fires when the user selects an Event Category Group. Users can select an Event Category Group by either clicking the Event Category Group with the mouse or by navigating through the TreeView using the keyboard cursor keys.

### **EventCategoryGroupID**

The ID of the newly selected Event Category Group.

### **EventCategorySelected**

```
public event EventCategorySelected(int EventCategoryID)
```

This event fires when the user selects an Event Category. Users can select an Event Category by either clicking the Event Category with the mouse or by navigating through the TreeView using the keyboard cursor keys.

### **EventCategoryID**

The ID of the newly selected Event Category.

## **Event Data Control**

The Event Data control provides programmatic access to the various properties of an Event. The control has no visual interface and is analogous to an object library rather than a visual component.

The Event Data control can load any Event and provide complete access to all of its properties.

## **Properties**

### **AcknowledgeDurationMinutes**

```
public int AcknowledgeDurationMinutes { get; }
```

Returns the amount of time (in Minutes) that a user has to acknowledge the Event after the Event ends and before the Event List control automatically acknowledges it. If the Event is not loaded, or if the Event is not configured to be broadcasted to the local machine, the result will be -1.

### **Active**

```
public int Active { get; }
```

Returns whether or not the loaded Event is active.

### **AllowAcknowledgeWhileActive**

```
public bool AllowAcknowledgeWhileActive { get; }
```

Returns whether or not the configuration allows users to acknowledge the Event while it is active. If the Event is not loaded, or if the Event is not configured to be broadcasted to the local machine, the result will be False.

## AreaID

```
public int AreaID { get; }
```

Returns the ID of the Area for the loaded Event.

## AreaName

```
public string AreaName { get; }
```

Returns the Name of the Area for the loaded Event.

## Capture\*

```
public string Capture01 { get; set; }
```

Gets or sets the appropriate Capture field of the loaded Event. There are 10 versions of this property, representing the 10 available Capture columns ranging from 01 to 10. If the Event is active, then these values cannot be changed. EVENTTrak will only save the licensed Capture fields to the database.

## Count

```
public int Count { get; set; }
```

Gets or sets the number of Events the loaded Event represents. When LOGICTrak creates an Event, it sets this value to 1, which represents a single Event. This primary purpose of this field is to support manual Event entry, where duration and time are unimportant but the number of occurrences is important.

## Data

```
public string Data { get; }
```

Returns the custom additional data LOGICTrak sent to EVENTTrak for the loaded Event. If the Event is not loaded, or if the Event is not configured to be broadcasted to the local machine, the result will be an empty string.

## DisplayDelaySeconds

```
public int DisplayDelaySeconds { get; }
```

Returns the minimum number of seconds that an Event must remain active before the Event List control displays the Event. If the Event is not loaded, or if the Event is not configured to be broadcasted to the local machine, the result will be 0.

## DurationSeconds

```
public int DurationSeconds { get; }
```

Returns the total number of seconds that the loaded Event has been active. If the Event is still Active, EVENTTrak will only update this number when it reloads the Event (using the LoadByID method).

## EndTime

```
public string EndDate { get; set; }
```

Gets or sets the End Date/Time of the loaded Event. If the Event is active, the result will be an empty string and will remain unchangeable while the Event is active.

## EventCategory\*ID

```
public int EventCategory01ID { get; set; }
```

Gets or sets the appropriate Event Category ID field of the loaded Event. There are 10 versions of this property, representing the 10 available Event Category columns ranging from 01 to 10. EVENTTrak will only save the licensed Event Category fields to the database.

## EventCategory\*Name

```
public string EventCategory01Name { get; }
```

Returns the appropriate Event Category Name field of the loaded Event. There are 10 versions of this property, representing the 10 available Event Category columns ranging from 01 to 10.

## EventCode

```
public string EventCode { get; }
```

Returns the Event Code's Code property for the loaded Event.

## EventCodeID

```
public int EventCodeID { get; set; }
```

Gets or sets the ID of the Event Code for the loaded Event.

## EventCodeName

```
public string EventCodeName { get; }
```

Returns the Name of the Event Code for the loaded Event.

## EventDate

```
public string EventDate { get; set; }
```

Gets or Sets the Event Date of the loaded Event. This value is read-only while the Event is active.

## EventDefinitionGroupName

```
public string EventDefinitionGroupName { get; }
```

Returns the Name of the Event Definition Group for the loaded Event.

## EventDefinitionID

```
public int EventDefinitionID { get; set; }
```

Gets or sets the ID of the Event Definition for the loaded Event. Changing this value will effectively change many properties of this control, including SystemID and AreaID. If you save the Event after modifying this property, it is recommended that you reload the Event (using the LoadByID method) before reusing any of its properties.

## EventDefinitionName

```
public string EventDefinitionName { get; }
```

Returns the Name of the Event Definition for the loaded Event.

## EventID

```
public int EventID { get; }
```

Returns the ID of the loaded Event.

## ForceAcknowledgeOnEventEnd

```
public bool ForceAcknowledgeOnEventEnd { get; }
```

Returns whether or not the Event List Control will automatically acknowledge an Event when the Event ends. If the Event is not loaded or the Event is not configured to be broadcasted to the local machine, the result will be false.

## Impact

```
public double Impact { get; set; }
```

Gets or sets the Impact of the loaded Event. The Impact represents the percentage of time the total duration of the Event actually impacted the System. For a normal Event, this corresponds to a value of 1.0 (the full duration). To use only half of the Event's time, set this value to 0.5. The sample application uses this property to split an Event.

## IsEdited

```
public bool IsEdited { get; set; }
```

Gets or sets whether or not the user has edited the loaded Event. For Events generated by the LOGICTrak, this value is False; for Events created by the WEBTrak Knowledge Management Portal Editor, this value is True.

## IsEventCategoryRequired

```
public bool IsEventCategoryRequired { get; }
```

Returns whether or not the Event is configured to allow a user to save or acknowledge an Event if an EventCategory has not been picked. If the Event is not



loaded or the Event isn't configured to be broadcasted to the machine, the result will be False.

### Notes

```
public string Notes { get; set; }
```

Gets or sets the freeform Notes for the loaded Event.

### OeeEventType

```
public int OeeEventType { get; set; }
```

Gets or sets the OEE Type of the loaded Event.

### OeeEventTypeName

```
public string OeeEventTypeName { get; }
```

Returns the Name of the OEE Type for the loaded Event.

### ProductCode

```
public string ProductCode { get; }
```

Returns the Product Code for the loaded Event.

### ProductID

```
public int ProductID { get; set; }
```

Gets or sets ID of the Product for the loaded Event.

### ProductName readonly

```
public string ProductName { get; }
```

Returns the Name of the Product for the loaded Event.

### ReadOnly

```
public bool ReadOnly { get; }
```

Returns whether or not the Event is configured to allow users to save or acknowledge the Event. If the Event is not loaded or the Event is not configured to be broadcasted to the local machine, the result will be False.

### ShiftHistoryID

```
public int ShiftHistoryID { get; set; }
```

Gets or sets the ShiftHistoryID of the loaded Event. The ShiftHistory represents a specific Shift, Team, Date (without time), and duration of a Shift. For example, a Shift might be "Morning Shift", whereas a ShiftHistory instance might be "Morning Shift" on September 18<sup>th</sup>, 2005 from 8:00 AM to 5:00 PM.

### **ShiftID**

```
public int ShiftID { get; }
```

Returns the ID of the Shift for the loaded Event.

### **ShiftName**

```
public string ShiftName { get; }
```

Returns the Name of the Shift for the loaded Event.

### **SplitEventID**

```
public int SplitEventID { get; set; }
```

Gets or sets the SplitEventID of the loaded Event. The SplitEventID represents a link to an Event if the Event was split. This value is read-only while the Event is active.

### **StartDateTime**

```
public string StartDateTime { get; set; }
```

Gets or sets the Start Date/Time of the loaded Event.

### **State**

```
public int State { get; set; }
```

Gets or sets the State of the loaded Event.

### **StateDateTime**

```
public string StateDateTime { get; set; }
```

Gets or sets the State Date/Time of the loaded Event.

### **StateName**

```
public string StateName { get; }
```

Returns the Name of the State for the loaded Event.

### **SubSystemID**

```
public int SubSystemID { get; }
```

Returns the ID of the Sub-System for the loaded Event.

### **SubSystemName**

```
public string SubSystemName { get; }
```

Returns the Name of the Sub-System for the loaded Event.

## **SystemID**

```
public int SystemID { get; }
```

Returns the ID of the System for the loaded Event.

## **SystemName**

```
public string SystemName { get; }
```

Returns the Name of the System for the loaded Event.

## **TeamID**

```
public int TeamID { get; }
```

Returns the ID of the Team for the loaded Event.

## **TeamName**

```
public string TeamName { get; }
```

Returns the Name of the Team for the loaded Event.

## **JobID**

```
public int JobID { get; set; }
```

Gets or sets the ID of the Job for the loaded Event.

## **JobName**

```
public string JobName { get; }
```

Returns the Name of the Job for the loaded Event.

## **BatchID**

```
public int BatchID { get; set; }
```

Gets or sets the ID of the Batch for the loaded Event.

## **BatchName**

```
public string BatchName { get; }
```

Returns the Name of the Batch for the loaded Event.

## **Methods**

### **Acknowledge**

```
public bool Acknowledge([bool IsAuto=False])
```

This method makes the necessary changes to the properties to acknowledge the Event based on the supplied parameter. In addition, it will clear any broadcast messages for this Event to other EVENTTrak nodes.

### **IsAuto**

This specifies whether the acknowledgement is an automatic acknowledgement or a user-initiated acknowledgement.

### **AcknowledgeWithAudit**

```
public bool AcknowledgeWithAudit([string User01], [string User02], [string Notes])
```

This method is the same as the Acknowledge() method but inserts the audit information passed in and associates it with the update. This method should only be used if AUDITTrak is licensed and enabled. The return value is False if the save fails.

### **User01 and User02**

These values are the primary and secondary signature users for the audit action.

### **Notes**

This is any notes that are to be associated with the audit action.

### **ClearFromEventMonitors**

```
public bool ClearFromEventMonitors()
```

This method clears all broadcast messages for the loaded Event, effectively removing the Event from any other EVENTTrak nodes.

### **FormatDuration**

```
public string FormatDuration([int Seconds], [string FormatString])
```

This method is primarily a utility method for formatting seconds in various formats. It uses the EVENTTrak Time Span Format.

### **Seconds**

This is the source number of seconds to format.

### **FormatString**

This string determines how to format the specified number of Seconds. For more information on the different formats available, see the section on the Time Span Format.

### **GetSelectedEventCategoryID**

```
public int GetSelectedEventCategoryID()
```

This method provides simple access to the “lowest” level Event Category selected. For example, if the user has selected a level 3 Event Category, this method will return the ID of the Event Category for that level.

## GetSelectedEventCategoryName

```
public string GetSelectedEventCategoryName()
```

This method provides simple access to the Name of the “lowest” level Event Category selected. For example, if the user has selected a level 3 Event Category, this method will return the Name of the Event Category for that level.

## LoadByID

```
public bool LoadByID([int EventID])
```

This method loads the control with all of the data for an Event. This is the starting point; no other methods (except FormatDuration) will work without a loaded Event.

## EventID

This is the ID of the Event to be load.

## Save

```
public bool Save()
```

This method saves any modifications to the loaded Event to the database.

## SaveAsNew

```
public bool SaveAsNew()
```

This method saves a new copy of the loaded Event. The control will generate a EventID, but all other properties will remain the same. This method helps facilitate splitting Events.

## SaveAsNewWithAudit

```
public bool SaveAsNewWithAudit([string User01], [string User02], [string Notes])
```

This method is the same as the SaveAsNew() method but inserts the audit information passed in and associates it with the update. This method should only be used if AUDITtrak is licensed and enabled. The return value is False if the save fails.

## User01 and User02

These values are the primary and secondary signature users for the audit action.

## Notes

This is any notes that are to be associated with the audit action.

## SaveWithAudit

```
public bool SaveWithAudit([string User01], [string User02], [string Notes])
```

This method is the same as the Save() method but inserts the audit information passed in and associates it with the update. This method should only be used if AUDITtrak is licensed and enabled. The return value is False if the save fails.

## **User01 and User02**

These values are the primary and secondary signature users for the audit action.

## **Notes**

This is any notes that are to be associated with the audit action.

## **SetEventCategories**

```
public bool SetEventCategories([int EventCategoryID])
```

This method is a utility method that provides a simple means to set all the properties based on a single Event Category's ID. This will set all of the necessary Event Category ID and Name properties, as well as any applicable Event Code and OeeType properties. This method does not automatically save these values to the database; it only changes the properties loaded in the control. You must call the Save method to save these changes to the database.

## **EventCategoryID**

This is the ID of the "lowest" level Event Category selected. The "lowest" possible Event Category level is 10.

## **SetPropertiesFromDbConfig**

```
public bool SetPropertiesFromDbConfig(DbConfig ConfigSettings)
```

This method will configure all of the control's database connection properties in one step via a reusable DbConfig object.

## **ConfigSettings**

The DbConfig object to use to configure the database connection properties for this control.

## **ValidateAuditorLogin**

```
public bool ValidateAuditorLogin ([string Login], [string Password])
```

This method validates a login and password based on the Authentication Type set in the TrakSYS settings. The return value is True if the login and password is valid and the user is configured as an Auditor.

## **Login**

This is the login as specified in a TrakSYS user record. For Windows Users, the login should include the domain in the format DOMAIN\login.

## **Password**

This is the TrakSYS or Windows password for the Login passed in. The password is case sensitive.

## **ValidateWindowsLogin**

```
public bool ValidateWindowsLogin ([string Login], [string Password])
```

This method validates a Windows login and password. The return value is True if the login exists in the specified domain and the password is correct.

### **Login**

This is a Windows login including the domain in the format DOMAIN\login.

### **Password**

This is the Windows password for the Login passed in. The password is case sensitive.

## **Event List Control**

The Event List control is the primary control for displaying and reacting to Events generated by LOGICTrak. You can use the control as a visible control to display up to 10 columns of Event data or as an invisible control in conjunction with custom code to react to various Events.

You must first set the various database connection properties and call the StartPolling method before using the control. For greater control over updates to the control's data, you can call PollData manually instead of using StartPolling for the automatic polling functionality.

## **Properties**

### **ActiveEventForeColor**

```
public int ActiveEventForeColor { get; set; }
```

Gets or Sets the color of the text the control uses to display active Events. This color only applies to active Events; once an Event becomes inactive, it uses the InactiveEventForeColor property.

### **ActiveEventIsBold**

```
public bool ActiveEventIsBold { get; set; }
```

Gets or sets whether or not the control should use bold text to display active Events. Once an Event becomes inactive, it uses the InactiveEventIsBold property.

### **BackColor**

```
public int BackColor { get; set; }
```

Gets or sets the background color for display text and graphics in the control.

### **Column\*Alignment**

```
public int Column1Alignment { get; set; }
```

Gets or sets how the control aligns column text. The following are the only valid values:

- 0 – Left Alignment
- 1 – Right Alignment

- 2 – Center Alignment

This property represents the alignment for a particular column. There are 10 versions of this property, representing the 10 columns which range from 0 to 9. Column0Alignment exists for consistency, but can only have the value of 0 (Left Alignment).

### **Column\*Caption**

```
public string Column0Caption { get; set; }
```

Gets or sets the display text for a particular Column Header at the top of the control. There are 10 versions of this property, representing the 10 columns which range from 0 to 9.

### **Column\*DataField**

```
public string Column0DataField { get; set; }
```

Gets or sets which Event value the control will display for the specified column. The following are valid values for this property:

- AreaName
- AreaName+SystemName
- Capture01
- Capture02
- Capture03
- Capture04
- Capture05
- Capture06
- Capture07
- Capture08
- Capture09
- Capture10
- Date
- DurationSeconds
- DurationSinceEndedSeconds
- DurationUntilAutoAcknowledgeSeconds
- EndDateTime
- EventCategory01Name
- EventCategory02Name
- EventCategory03Name
- EventCategory04Name
- EventCategory05Name
- EventCategory06Name
- EventCategory07Name
- EventCategory08Name
- EventCategory09Name
- EventCategory10Name
- EventCode
- EventCodeName
- EventDefinitionGroupName
- EventDefinitionGroupName+EventDefinitionName
- EventDefinitionName



- OeeEventTypeName
- ProductCode
- ProductName
- ShiftName
- ShiftName+TeamName
- StartDateTime
- StateDateTime
- StateName
- SubSystemName
- SubSystemName+EventDefinitionName
- SystemName
- SystemName+EventDefinitionName
- SystemName+SubSystemName
- TeamName
- TeamName+ShiftName

There are 10 versions of this property, representing the 10 columns which range from 0 to 9.

### **Column\*Format**

```
public string Column0Format { get; set; }
```

Gets or sets how the control will display the value of the Column\*DataField. By default, this property is an empty string which uses the default format. For the majority of values specified in the Column\*DataField property, standard Microsoft Visual Basic 6.0 formatting can be applied (the formatting is similar to Microsoft Excel custom formatting). For reference see <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbentlr98/html/vafctFormat.asp>. For the following Column\*DataField property values, use the custom Time Span formatting:

- DurationSeconds
- DurationSinceEndedSeconds
- DurationUntilAutoAcknowledgeSeconds

There are 10 versions of this property, representing the 10 columns which range from 0 to 9.

### **Column\*Width**

```
public float Column0Width { get; set; }
```

Gets or sets the width of a Column. If a column should be hidden, set the width to 0. There are 10 versions of this property, representing the 10 columns which range from 0 to 9.

### **EventCount**

```
public int EventCount { get; }
```

Returns the total number of Events that are in the control.

## Font

```
public Font Font { get; set; }
```

Gets or sets the font the control uses to display text in the control. This is a standard object which provides access to every available font property. If the ActiveX container does not support objects, use the Font properties below to adjust the font for the control.

## FontBold

```
public bool FontBold { get; set; }
```

Gets or sets whether or not the control's text will appear bold.

## FontCharset

```
public int FontCharset { get; set; }
```

Gets or sets the character set the control uses to display text. This property primarily helps support various language settings.

## FontName

```
public string FontName { get; set; }
```

Gets or sets the name of the font the control uses to display text.

## FontSize

```
public float FontSize { get; set; }
```

Gets or sets the font size the control uses to display text.

## GridLines

```
public bool GridLines { get; set; }
```

Gets or sets whether or not grid lines appear between rows and columns.

## HideColumnHeaders

```
public bool HideColumnHeaders { get; set; }
```

Gets or sets whether or not the control's column headers are hidden or visible.

## ImageType

```
public int ImageType { get; set; }
```

Gets or sets the type of icon to display. The following are the only valid values:

- 0 – None
- 1 – Small Image
- 2 – Large Image

If the value is 0, there will be no image. If the value is 1, the control will use a 16 x 16 pixel image. If the value is 2, the control will use a 32 x 32 pixel image. There are separate images for active Events and inactive Events.

### **InactiveEventForeColor**

```
public int InactiveEventForeColor { get; set; }
```

Gets or Sets the color of the text the control uses to display inactive Events. Active Events use the ActiveEventForeColor property instead.

### **InactiveEventIsBold**

```
public bool InactiveEventIsBold { get; set; }
```

Gets or sets whether or not the text the control use to display inactive Events is bold. Active Events use the ActiveEventIsBold property instead.

### **IsDbConnected**

```
public bool IsDbConnected { get; }
```

Returns whether or not the control has successfully connected to the SQL Server database.

### **IsPolling**

```
public bool IsPolling { get; }
```

Returns whether or not the control is actively polling the SQL Server database. If this value is false, the control is not automatically refreshing its data. To change the value of this property, call the StartPolling or StopPolling methods.

### **PollIntervalSeconds**

```
public int PollIntervalSeconds { get; set; }
```

Gets or sets the time (in seconds) between automatic calls to the PollData method when the control is actively polling.

### **SelectedIndex**

```
public int SelectedIndex { get; set; }
```

Gets or sets the index for the selected row in the Event List. The index is 0-based.

### **SystemIDs**

```
public string SelectedIndex { get; set; }
```

Gets or sets a comma delimited list of System IDs. When this property is set to a non-empty string, the Event List will only display events for the specified System IDs. The default is an empty string which causes the Event List to display events for all Systems that are configured for the specified node.

## Methods

### GetDbDateTime

```
public string GetDbDateTime([string TimeFormat=""])
```

This returns the value of the current Date/Time on the SQL Server host. This clock is the basis for all TrakSYS operations.

#### TimeFormat

This parameter is a string specifying how to format the Date/Time as a string. The format uses standard Microsoft Visual Basic formatting.

### GetColumnValueByIndex

```
public string GetColumnValueByIndex([int Index=-1], [int Column=0])
```

This returns the value of specified column for the specified row (index) for the Event List control. The value returned is a string and is the exact value that is displayed in the Event List control.

#### Index

This parameter is the 0-based index which specifies which row in the Event List control to access.

#### Column

This parameter is the 0-based column number which specifies which column in the Event List control to access.

### GetIndexByEventID

```
public int GetIndexByEventID([int EventID])
```

This returns the 0-based Index based on the passed in parameter.

#### EventID

This is the ID of an Event currently loaded in the control.

### GetEventIDByIndex

```
public int GetEventIDByIndex([int Index])
```

This returns the Event ID based on the specified index. Although the parameter is optional, it is only optional for compatibility with substandard Active X containers. If the Index parameter is omitted, this method will return a value of -1.

This method is provided to allow integration with the Event Data Control and to provide the capability to loop through a list of all Events in the control (in conjunction with the EventCount property).

#### Index

This is the 0-based index of the appropriate Event in the control.

## GetSelectedEventID

```
public int GetSelectedEventID()
```

This returns the ID of the selected Event in the control. If there are no events in the control, the result will be -1.

## PollData

```
public void PollData()
```

This method is the main process of the Event List control. This performs all logic for acknowledging and displaying Events. The control automatically calls this method while it is actively polling (IsPolling is True). In addition, this method raises 2 events, BeforePollData and AfterPollData, allowing the ActiveX container to perform any necessary pre-poll or post-poll processing.

## SetPropertiesFromDbConfig

```
public bool SetPropertiesFromDbConfig(DbConfig ConfigSettings)
```

This method will configure all of the control's database connection properties in one step via a reusable DbConfig object.

## ConfigSettings

The DbConfig object to use to configure the database connection properties for this control.

## StartPolling

```
public void StartPolling()
```

This method attempts to begin actively polling the database. This uses the PollIntervalSeconds property to determine the time between automatic calls to the PollData method.

## StopPolling

```
public void StopPolling()
```

This method will cease any ongoing, automatic polling.

## Events

### AfterPollData

```
public event AfterPollData()
```

This event fires after the PollData method completes, regardless of whether the call to PollData occurs automatically or manually.

## **BeforePollData**

```
public event BeforePollData()
```

This event fires before the PollData method completes all of its processing, regardless of whether the call to PollData occurs automatically or manually.

## **DbDisconnected**

```
public event DbDisconnected()
```

This event fires if the control encounters an error attempting to establish a connection to the SQL Server database. It allows the programmer to react to database connectivity errors without constantly checking the IsDbConnected property value.

## **DbReconnected**

```
public event DbReconnected()
```

This event fires if the control encounters an error attempting to connect to the SQL Server database and then successfully reconnects. This event will fire if the control successfully reconnects during the current its lifespan. If the container application repeatedly loads and unloads the control, this event may never fire.

## **EventAutoPosted**

```
Public event EventAutoPosted(int EventID)
```

This event fires when an Event has exceeded the allotted time for a user response and is being automatically acknowledged by the Event List control.

### **EventID**

This is the ID of the Event automatically acknowledged.

## **EventClick**

```
public event EventClick(int EventID)
```

This event fires when the user clicks an Event in the control. This occurs regardless of whether the click is a left or right mouse click, but only if the user clicks an Event.

### **EventID**

This is the ID of the Event clicked by the user.

## **EventDisplayed**

```
public event EventDisplayed(int EventID)
```

This event fires when the control displays an active Event for the first time.

### **EventID**

This is the ID of the newly displayed Event.

## EventDoubleClick

```
public event EventDoubleClick(int EventID)
```

This event fires when the user double-clicks an Event in the control. This occurs regardless of whether the double-click is a left or right mouse click, but only if the user double-clicks an Event.

### EventID

This is the ID of the Event double-clicked by the user.

## EventEnded

```
public event EventEnded(int EventID)
```

This event fires when a displayed Event ends (becomes inactive).

### EventID

This is the ID of the ended Event.

## EventRemoved

```
public event EventRemoved(int EventID)
```

This event fires when an Event displayed in the control is no longer configured to be displayed. This usually occurs when an EVENTTrak on another machine acknowledges an Event.

### EventID

This is the ID of the removed Event.

## EventSelected

```
public event EventSelected(int EventID)
```

This event fires when the user selects an Event. The user can select an Event by clicking it with the mouse or by navigating through the list using the keyboard cursor keys.

### EventID

This is the ID of the newly selected Event.

## Oee Data Control

The Oee Data control provides programmatic access to OEE data. The control has no visual interface and is analogous to an object library rather than a visual component.

The Oee Data control can capture the various Key Performance Indicators configured for a KPI Calculation using the PollData method, or continuously update them using the automatic polling mechanisms provided by the StartPolling and StopPolling methods.

## Properties

## Availability

```
public double Availability { get; }
```

Returns the Availability based on the filtered criteria as a percentage (where 50% is 50 not 0.5).

## AvailabilityIsNaN

```
public bool AvailabilityIsNaN { get; }
```

Returns whether or not the Availability result is not a number (i.e. NaN). This is possible if the calculation for Availability results in a division by zero.

## AvailabilityLossSeconds

```
public int AvailabilityLossSeconds { get; }
```

Returns the total Availability Loss in seconds that occurred during the current filtered criteria.

## BadCalculationUnitsAdjusted

```
public int BadCalculationUnitsAdjusted { get; }
```

Returns the number of Bad Calculation Units for the current filtered criteria. This number includes the initial number of Bad Units captured, as well as any additional units applied as adjustments. The units are set according to the KPI Calculation's Calculation Units.

## BadCalculationUnitsInitial

```
public int BadCalculationUnitsInitial { get; }
```

Returns the number of Bad CalculationUnits for the current filtered criteria. This number does not include any of the adjustments that have been applied. The units are set based on the KPI Calculation's Calculation Units.

## BadDisplayUnitsAdjusted

```
public int BadDisplayUnitsAdjusted { get; }
```

Returns the number of Bad Display Units for the current filtered criteria. This number includes the initial number of Bad Units captured, as well as any additional units applied as adjustments. The control automatically converts the units to Display Units.

## BadDisplayUnitsInitial

```
public int BadDisplayUnitsInitial { get; }
```

Returns the number of Bad Display Units for the current filtered criteria. This number does not include any adjustments. The control automatically converts the units to Display Units.



## **BadDisplayUnitsName**

```
public string BadDisplayUnitsName { get; }
```

Returns the name of the Engineering Units that represent the Bad Display Units.

## **BaselineAvailability**

```
public double BaselineAvailability { get; }
```

Returns the Baseline Availability percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## **BaselineOee**

```
public double BaselineOee { get; }
```

Returns the Baseline OEE percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## **BaselinePerformance**

```
public double Baseline Performance { get; }
```

Returns the Baseline Performance percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## **BaselineQuality**

```
public double BaselineQuality { get; }
```

Returns the Baseline Quality percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## **BaselineTeep**

```
public double BaselineTeep { get; }
```

Returns the Baseline TEEP percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## **CalculationUnitsName**

```
public string CalculationUnitsName { get; }
```

Returns the name of the Engineering Units that represent the Calculation Units.

## **Capture\***

```
public string Capture01 { get; set; }
```

Gets or sets the appropriate Capture field to filter the retrieval of the OEE data. There are 10 versions of this property, representing the 10 available Capture columns ranging from 01 to 10. Only the licensed Capture fields will affect the data filtering.

## EndTime

```
public string EndDateTime { get; set; }
```

Gets or sets the End Date/Time to filter the retrieval of the OEE data. This is one of the required properties that must be set in order for the PollData method to successfully retrieve the data. If the UseCurrentShiftHistoryID property is True, the control will set the EndDateTime property automatically; otherwise, set this property during the BeforePollData event.

## GoodCalculationUnitsAdjusted

```
public int GoodCalculationUnitsAdjusted { get; }
```

Returns the number of Good Calculation Units for the current filtered criteria. This number includes the initial number of Good Units captured, as well as any additional units applied as adjustments. The units are set according to the KPI Calculation's Calculation Units.

## GoodCalculationUnitsInitial

```
public int GoodCalculationUnitsInitial { get; }
```

Returns the number of Good Calculation Units for the current filtered criteria. This number does not include any adjustments. The units are set according to the KPI Calculation's Calculation Units.

## GoodDisplayUnitsAdjusted

```
public int GoodDisplayUnitsAdjusted { get; }
```

Returns the number of Good Display Units for the current filtered criteria. This number includes the initial number of Good Units captured, as well as any additional units applied as adjustments. The control automatically converts the units to Display Units.

## GoodDisplayUnitsInitial

```
public int GoodDisplayUnitsInitial { get; }
```

Returns the number of Good Display Units for the current filtered criteria. This number does not include any adjustments. The control automatically converts the units to Display Units.

## GoodDisplayUnitsName

```
public string GoodDisplayUnitsName { get; }
```

Returns the name of the Engineering Units that represent the Good Display Units.

## IncludeExceptions

```
public bool IncludeExceptions { get; set; }
```

Gets or sets whether or not the current filtered criteria will include any KPI Intervals marked as Exceptions.

### **IntervalEndTime**

```
public string IntervalEndTime { get; }
```

Returns the End Date/Time of the current filtered criteria. Although an End Date/Time is part of the filter criteria, the actual End Date/Time of the retrieved data may be before that Date/Time.

### **IsDbConnected**

```
public bool IsDbConnected { get; }
```

Returns whether or not the control has successfully connected to the SQL Server database.

### **IsPolling**

```
public bool IsPolling { get; }
```

Returns whether or not the control is actively polling the SQL Server database. If this value is false, then the control is not automatically refreshing the data in the control. To change the value of this property, call the StartPolling or StopPolling methods.

### **NaNValue**

```
public double NaNValue { get; set; }
```

Gets or sets the value to use as NaN (not a number) for the various Key Performance Indicators. If any of the data elements experience a division by zero error, the control will use the value of this property.

### **NetOperationSeconds**

```
public int NetOperationSeconds { get; }
```

Returns the Total Seconds of the current filter criteria minus the Total System Not Scheduled Seconds.

### **Oee**

```
public double Oee { get; }
```

Returns the OEE based on the filtered criteria as a percentage (where 50% is 50 not 0.5).

### **OeeCalculationID**

```
public int OeeCalculationID { get; set; }
```

Gets or sets the KPI Calculation ID to filter the retrieval of the OEE data. This is one of the required properties that must be set in order for the PollData method to successfully retrieve data.

### **OeeCalculationName**

```
public string OeeCalculationName { get; }
```

Returns the Name of the KPI Calculation being filtered.

### **OeeIsNaN**

```
public bool OeeIsNaN { get; }
```

Returns whether or not the OEE result is not a number (NaN). This is possible if the calculation for OEE results in a division by zero.

### **Performance**

```
public double Performance { get; }
```

Returns the calculated Performance based on the filter criteria as a percentage (where 50% is 50 not 0.5).

### **PerformanceIsNaN**

```
public bool PerformanceIsNaN { get; }
```

Returns whether or not the Performance result is not a number (NaN). This is possible if the calculation for Performance results in a division by zero.

### **PollIntervalMinutes**

```
public int PollIntervalMinutes { get; set; }
```

Gets or sets the time (in minutes) between automatic calls to the PollData method when the control is actively polling. The interval is in minutes due to the way TrakSYS stores OEE data. OEE is typically only stored once or twice an hour; as a result, this control does not need to update very frequently.

### **ProductCode**

```
public string ProductCode { get; }
```

Returns the Product Code if the filter criteria includes a Product.

### **ProductID**

```
public int ProductID { get; set; }
```

Gets or sets the Product ID to filter the retrieval of the OEE data.

### **ProductionSeconds**

```
public int ProductionSeconds { get; }
```

Returns the Total Seconds of the current filter criteria minus the Total System Not Scheduled Seconds and the Availability Loss Seconds.

### **ProductName**

```
public string ProductName { get; }
```

Returns the Product Name if the filter criteria includes a Product.

### **Quality**

```
public double Quality { get; }
```

Returns the Quality based on the filter criteria as a percentage (where 50% is 50 not 0.5).

### **QualityIsNaN**

```
public bool QualityIsNaN { get; }
```

Returns whether or not the Quality result is not a number (NaN). This is possible if the calculation for Quality results in a division by zero.

### **ShiftHistoryID**

```
public int ShiftHistoryID { get; set; }
```

Gets or sets the ShiftHistoryID to filter the retrieval of the OEE data. A ShiftHistory represents an instance of a specific Shift, Team, Date (without time), and duration. For example, a Shift might be "Morning Shift", whereas a ShiftHistory might be "Morning Shift" on September 18<sup>th</sup>, 2005 from 8:00 am to 5:00 PM.

### **ShiftID**

```
public int ShiftID { get; set; }
```

Gets or sets the Shift ID to filter the retrieval of the OEE data.

### **ShiftName**

```
public string ShiftName { get; }
```

Returns the Shift Name if the filter criteria includes a Shift;

### **StartDateTime**

```
public string StartDateTime { get; set; }
```

Gets or sets the Start Date/Time to filter the retrieval of the OEE data. This is one of the required properties that must be set in order for the PollData method to successfully retrieve data. If the UseCurrentShiftHistoryID property is True, the control will set the StartDateTime property automatically; otherwise, set this property during the BeforePollData event.

## SystemNotScheduleSeconds

```
public int SystemNotScheduleSeconds { get; }
```

Returns the total System Not Scheduled in seconds that occurred during the current filter criteria.

## TaktSeconds

```
public double TaktSeconds { get; }
```

Returns the Takt Time based on the filter criteria.

## TaktSecondsIsNaN

```
public bool TaktSecondsIsNaN { get; }
```

Returns whether or not the Takt result is not a number (NaN). This is possible if the calculation for Takt Time results in a division by zero.

## TargetAvailability

```
public double TargetAvailability { get; }
```

Returns the Target Availability percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## TargetOee

```
public double TargetOee { get; }
```

Returns the Target OEE percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## TargetPerformance

```
public double TargetPerformance { get; }
```

Returns the Target Performance percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## TargetQuality

```
public double TargetQuality { get; }
```

Returns the Target Quality percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## TargetTeep

```
public double TargetTeep { get; }
```

Returns the Target TEEP percentage configured for the KPI Calculation as a percentage (where 50% is 50 not 0.5).

## TeamID

```
public int TeamID { get; set; }
```

Gets or sets the Team ID to filter the retrieval of the OEE data.

## TeamName

```
Public string TeamName { get; }
```

Returns the Team Name if the filter criteria includes a Team.

## Teep

```
public double Teep { get; }
```

Returns the TEEP based on the filter criteria as a percentage (where 50% is 50 not 0.5).

## TeepIsNaN

```
public bool TeepIsNaN { get; }
```

Returns whether or not the TEEP result is not a number (NaN). This is possible if the calculation for TEEP results in a division by zero.

## TotalCalculationUnitsAdjusted

```
public int TotalCalculationUnitsAdjusted { get; }
```

Returns the number of Total Calculation Units for the current filter criteria. This number includes the initial number of Total Units captured, as well as any additional units applied as adjustments. The units are set according to the KPI Calculation's Calculation Units.

## TotalCalculationUnitsInitial

```
public int TotalCalculationUnitsInitial { get; }
```

Returns the number of Total Calculation Units for the current filter criteria. This number does not include any adjustments. The units are set according to the KPI Calculation's Calculation Units.

## TotalDisplayUnitsAdjusted

```
public int TotalDisplayUnitsAdjusted { get; }
```

Returns the number of Total Display Units for the current filter criteria. This number includes the initial number of Total Units captured, as well as any additional units applied as adjustments. The control automatically converts the units to Display Units.

## TotalDisplayUnitsInitial

```
public int TotalDisplayUnitsInitial { get; }
```

Returns the number of Total Display Units for the current filter criteria. This number does not include any adjustments. The control automatically converts the units to Display Units.

### **TotalDisplayUnitsName**

```
public string TotalDisplayUnitsName { get; }
```

Returns the name of the Engineering Units that represent the Total Display Units.

### **TotalSeconds**

```
public int TotalSeconds { get; }
```

Returns the total Seconds that occurred during the current filter criteria.

### **UseCurrentCapture\***

```
public bool UseCurrentCapture01 { get; set; }
```

Gets or sets whether or not the control should retrieve, set, and filter data by the appropriate Capture field. The value of this is a wildcard search, with % being the wildcard character. For example, to include only data that starts with A for Capture01, the value should be "A%".

### **UseCurrentProductID**

```
public bool UseCurrentProductID { get; set; }
```

Gets or sets whether or not the control should retrieve, set, and filter data by the current Product ID.

### **UseCurrentShiftHistoryID**

```
public bool UseCurrentShiftHistoryID { get; set; }
```

Gets or sets whether or not the control should retrieve the current ShiftHistory should before retrieving OEE Data. If this property is True, the control will set the following fields using the current values:

- ShiftHistoryID
- ShiftID
- TeamID
- StartDateTime
- EndDateTime

### **UseCurrentShiftID**

```
public bool UseCurrentShiftID { get; set; }
```

Gets or sets whether or not the control should retrieve, set, and filter data by the current Shift ID. Note that if UseCurrentShiftHistoryID is set to True, the control will ignore this value.



## UseCurrentTeamID

```
public bool UseCurrentTeamID { get; set; }
```

Gets or sets whether or not the control should retrieve, set, and filter data by the current Team ID. Note that if the UseCurrentShiftHistoryID is set to True, the control will ignore this value.

## UsesCurrentFilter

```
public bool UsesCurrentFilter { get; }
```

Returns whether or not any of the properties are configured to use the current information to retrieve data.

## JobID

```
public int JobID { get; set; }
```

Gets or sets the ID of the Job for the retrieval of the OEE data.

## JobName

```
public string JobName { get; }
```

Returns the Name of the Job for OEE data being retrieved.

## BatchID

```
public int BatchID { get; set; }
```

Gets or sets the ID of the Batch for the loaded retrieval of the OEE data.

## BatchName

```
public string BatchName { get; }
```

Returns the Name of the Batch for OEE data being retrieved.

## UseCurrentJobID

```
public bool UseCurrentJobID { get; set; }
```

Gets or sets whether or not the control should retrieve, set, and filter data by the current Job ID.

## UseCurrentBatchID

```
public bool UseCurrentBatchID { get; set; }
```

Gets or sets whether or not the control should retrieve, set, and filter data by the current Batch ID.

## Methods

## FormatDuration

```
public string FormatDuration([int Seconds], [string FormatString])
```

This is primarily a utility method for formatting seconds in various formats. It uses the EVENTTrak Time Span Format.

### Seconds

This is the source number of seconds to format.

### FormatString

This string determines how to format the specified seconds as a string. For more information on the different formats available, see the section on Time Span Format.

## GetDbDateTime

```
public string GetDbDateTime([string TimeFormat=""])
```

This returns the value of the current Date/Time on the SQL Server host. This clock is the basis for all TrakSYS operations.

### TimeFormat

This parameter is a string value specifying how to format the Date/Time value as a string. It uses standard Microsoft Visual Basic formatting.

## LoadCurrentFilterData

```
public bool LoadCurrentFilterData()
```

This method performs the necessary retrieval of data for any of the properties marked to retrieve the current information. The PollData method calls this method prior to raising the BeforePollData event.

## PollData

```
public void PollData()
```

This method is the main process of the Oee Data control. It performs the retrieval of all OEE data. If any of the properties are using current information (i.e. UsesCurrentFilter is True), the control calls LoadCurrentFilterData prior to raising the BeforePollData event. Once the control has retrieved and calculated the OEE data, it raises the AfterPollData event to allow for user interface updates.

## StartPolling

```
public void StartPolling()
```

This method attempts to begin actively polling the database. This uses the PollIntervalMintues property to determine the time between automatic calls to the PollData method.

## StopPolling

```
public void StopPolling()
```

This method will cease any ongoing, automatic polling.

### **SetPropertiesFromDbConfig**

```
public bool SetPropertiesFromDbConfig(DbConfig ConfigSettings)
```

This method will configure all of the control's database connection properties in one step via a reusable DbConfig object.

### **ConfigSettings**

The DbConfig object to use to configure the database connection properties for this control.

## **Events**

### **AfterPollData**

```
public event AfterPollData()
```

This event fires after the PollData method completes, regardless of whether the call to PollData occurs automatically or manually.

### **BeforePollData**

```
public event BeforePollData()
```

This event fires before the PollData method begins some of its processing, regardless of whether the call to PollData occurs automatically or manually. If any filter criteria need to be changed before calculating the OEE data, you can change them during this Event.

### **DbDisconnected**

```
public event DbDisconnected()
```

This event fires if the control encounters an error attempting to establish a connection to the SQL Server database. It allows the programmer to react to database connectivity errors without constantly checking the IsDbConnected property value.

### **DbReconnected**

```
public event DbReconnected()
```

This event fires if the control encounters an error attempting to connect to the SQL Server database and then successfully reconnects. This event will fire if the control successfully reconnects during the current its lifespan. If the container application repeatedly loads and unloads the control, this event may never fire.

## Product Picker Control

The Product Picker control allows the user to select an appropriate Product. This provides simple access to the complex hierarchy of Products and the various Product Attributes.

### Properties

#### Attribute\*DataType

```
public int Attribute01DataType { get; }
```

Returns the appropriate Attribute Data Type reflecting the intended Data Type of the Product Attribute. There are 10 versions of this property, representing the 10 available Attribute properties ranging from 01 to 10.

#### Attribute\*DataTypeName

```
public string Attribute01DataTypeName { get; }
```

Returns the appropriate Attribute Data Type Name corresponding to the intended Data Type of the Product Attribute. There are 10 versions of this property, representing the 10 available Attribute properties ranging from 01 to 10.

#### Attribute\*Name

```
public string Attribute01Name { get; }
```

Returns the name of the Product Attribute. There are 10 versions of this property, representing the 10 available Attribute properties ranging from 01 to 10.

#### Attribute\*IsEnabled

```
public bool Attribute01IsEnabled { get; }
```

Returns whether or not the Product Attribute is Enabled. There are 10 versions of this property, representing the 10 available Attribute properties ranging from 01 to 10.

#### DefaultProductCode

```
public string DefaultProductCode { get; }
```

Returns the Code for the Default Product if the currently loaded Product Set has a Default Product.

#### DefaultProductID

```
public int DefaultProductID { get; }
```

Returns the ID of the Default Product if the currently loaded Product Set has a Default Product.

## DefaultProductName

```
public string DefaultProductName { get; }
```

Returns the Name of the Default Product if the currently loaded Product Set has a Default Product.

## Font

```
public Font Font { get; set; }
```

Gets or sets the font EVENTTrak uses to display text on the control. This is a standard object which provides access to every available font property. If the ActiveX container does not support objects, you can use the other Font properties to adjust the control's font.

## FontBold

```
public bool FontBold { get; set; }
```

Gets or sets whether or not the control's text will appear bold.

## FontCharset

```
public int FontCharset { get; set; }
```

Gets or sets the character set the control uses to display text. This property primarily helps support various language settings.

## FontName

```
public string FontName { get; set; }
```

Gets or sets the name of the font the control uses to display text.

## FontSize

```
public float FontSize { get; set; }
```

Gets or sets the font size the control uses to display text.

## ForeColor

```
public int ForeColor { get; set; }
```

Gets or sets the color the control uses to display text.

## ImageType

```
public int ImageType { get; set; }
```

Gets or sets the type of icon to display. The following are the only valid values:

- 0 – None
- 1 – Small Image
- 2 – Large Image

If the value is 0, there will be no image. If the value is 1, the control will use a 16 x 16 pixel image. If the value is 2, the control will use a 32 x 32 pixel image. There are separate images for Products and Product Groups.

### **ProductCount**

```
public int ProductCount { get; }
```

Returns the total number of Products that are currently loaded in the control.

### **ProductGroupCount**

```
public int ProductGroupCount { get; }
```

Returns the total number of Product Groups that are currently loaded in the control.

### **ProductGroupTerm**

```
public string ProductGroupTerm { get; }
```

Returns the value that a Product Group is referred to for the currently loaded Product Set.

### **ProductSchemeID**

```
public int ProductSchemeID { get; }
```

Returns the Product Scheme ID for the currently loaded Product Set.

### **ProductSetDescription**

```
public string ProductSetDescription { get; }
```

Returns the Description for the currently loaded Product Set.

### **ProductSetID**

```
public int ProductSetID { get; }
```

Returns the ID for the currently loaded Product Set.

### **ProductSetName**

```
public string ProductSetName { get; }
```

Returns the Name for the currently loaded Product Set.

### **ProductSetNotes**

```
public string ProductSetNotes { get; }
```

Returns the Notes for the currently loaded Product Set.

## ProductSetTerm

```
public string ProductSetTerm { get; }
```

Returns the Term for the currently loaded Product Set.

## ProductTerm

```
public string ProductTerm { get; }
```

Returns the value that a Product is referred to for the currently loaded ProductSet.

## SelectedGroupDescription

```
public string SelectedGroupDescription { get; }
```

Returns the Description of the selected Product Group, or the Description of the selected Product's Product Group, if applicable.

## SelectedGroupID

```
public int SelectedGroupID { get; }
```

Returns the ID of the selected Product Group, or the ID of the selected Product's Product Group, if applicable.

## SelectedGroupName

```
public string SelectedGroupName { get; }
```

Returns the Name of the selected Product Group, or the Name of the selected Product's Product Group, if applicable.

## SelectedGroupParentGroupID

```
public int SelectedGroupParentGroupID { get; }
```

Returns the ParentGroupID of the selected Product Group, or the ParentGroupID for the selected Product's Product Group, if applicable.

## SelectedProductAttribute\*

```
public string SelectedProductAttribute01 { get; }
```

Returns the appropriate Attribute for the selected Product if one is selected and the Attribute is Enabled.

## SelectedProductCode

```
public string SelectedProductCode { get; set; }
```

Gets or sets the selected Product's Code.

### **SelectedProductDescription**

```
public string SelectedProductDescription { get; }
```

Returns the Description for the selected Product.

### **SelectedProductID**

```
public int SelectedProductID { get; set; }
```

Gets or sets the ID of the selected Product.

### **SelectedProductName**

```
public string SelectedProductName { get; }
```

Returns the Name for the selected Product.

### **SelectedProductNotes**

```
public string SelectedProductNotes { get; }
```

Returns the Notes for the Selected Product.

### **ShowProductsAfterGroups**

```
public bool ShowProductsAfterGroups { get; set; }
```

Gets or sets whether to display the Products after or before the Product Groups.

## **Methods**

### **CollapseAll**

```
public void CollapseAll()
```

This method will collapse all the TreeView nodes, leaving nothing expanded.

### **CollapseProduct**

```
public void CollapseProduct([int ProductID])
```

This method will collapse the TreeView node for the corresponding Product.

#### **ProductID**

This is the ID of the Product to collapse.

### **CollapseProductGroup**

```
public void CollapseProductGroup([int ProductGroupID])
```

This method will collapse the TreeView node for the corresponding Product Group.



**ProductGroupID**

This is the ID of the Product Group to collapse.

**ExpandAll**

```
public void ExpandAll()
```

This method will expand all the TreeView nodes, leaving nothing collapsed.

**ExpandProduct**

```
public void ExpandProduct([int ProductID])
```

This method will expand the TreeView node for the corresponding Product.

**ProductID**

This is the ID of the Product to expand.

**ExpandProductGroup**

```
public void ExpandProductGroup([int ProductGroupID])
```

This method will expand the TreeView node for the corresponding Product Group.

**ProductGroupID**

This is the ID of the Product Group to expand.

**GetProductGroupIDByIndex**

```
public int GetProductGroupIDByIndex([int Index])
```

This method will return the Product Group ID according to the specified index. If you omit the Index parameter, this method will return a value of -1.

**Index**

This is the 0-based index of the appropriate item in the control.

**GetProductIDByIndex**

```
public int GetProductIDByIndex([int Index])
```

This method will return the Product ID according to the specified index. If you omit the Index parameter, this method will return a value of -1.

**Index**

This is the 0-based index of the appropriate item in the control.

**GetProductTagIDForSystemID**

```
public int GetProductTagIDForSystemID([int SystemID])
```

This method will return the Product Tag ID for the specified System ID. You can use this Tag ID in conjunction with the TagData or TagList control to monitor the current Product.

**SystemID**

The ID of a System.

**SetSelectedProductFromCurrentProduct**

```
public bool SetSelectedProductFromCurrentProduct([int SystemID])
```

This method sets the selected Product in the control according to the specified System's current Product.

**SystemID**

The ID of a System. The control will use the value of the System's Product Tag to select the corresponding Product.

**SetProductTagBySystemID**

```
public bool SetProductTagBySystemID ([int SystemID])
```

This method updates the Product Tag for the specified System with the currently selected Product in the control. This method is particularly useful if you want to allow users to change the current Product for a System.

**SystemID**

The ID of a System. The control will update the value of the System's Product Tag with the currently selected Product.

**SetProductTagByTagName**

```
public bool SetProductTagByTagName([string TagName])
```

This method updates the specified Product Tag with the currently selected Product in the control. Use this method to update any Product Tag in the configuration, independent of the Product Map or System.

**TagName**

The Name of a Product Tag. Changing the value of this Tag will change the current Product of any Product Map or System referencing it.

**SetProductTagByTagID**

```
public bool SetProductTagByTagID([int TagID])
```

This method updates the specified Product Tag with the currently selected Product in the control. Use this method to update any Product Tag in the configuration, independent of the Product Map or System.

**TagID**

The ID of a Product Tag. Changing the value of this Tag will change the current Product of any Product Map or System referencing it.

## LoadByProductSetID

```
public bool LoadByProductSetID(  
    [int ProductSetID],  
    [int SelectedProductID]  
)
```

This method will load all the Products and Product Groups for the specified Product Set. This will load all of the appropriate data based on the current configuration in the database.

### ProductSetID

This is the Product Set ID to load.

### SelectedProductID

This allows you to programmatically pre-select the Product by ID during the load process, which will also ensure that the selected Product is visible once the load is complete.

## LoadBySystemID

```
public bool LoadBySystemID(  
    [int SystemID],  
    [int SelectedProductID]  
)
```

This method will load all of the Products and Product Groups for the Product Set linked to the specified System. This will load all of the appropriate data based the current configuration in the database.

### SystemID

This is the System ID to load.

### SelectedProductID

This allows you to programmatically pre-select the Product by ID during the load process, which will also ensure that the selected Product is visible once the load is complete.

## MakeProductGroupVisible

```
public void MakeProductGroupVisible([int ProductGroupID])
```

This method will expand all necessary TreeView nodes so that the specified Product Group will be visible in the TreeView.

### ProductGroupID

This is the Product Group ID to be made Visible.

## MakeProductVisible

```
public void MakeProductVisible([int ProductID])
```

This method will expand all the necessary TreeView nodes so that the specified Product will be visible in the TreeView.

**ProductID**

This is the Product ID to be made Visible.

**SetPropertiesFromDbConfig**

```
public bool SetPropertiesFromDbConfig(DbConfig ConfigSettings)
```

This method will configure all of the control's database connection properties in one step via a reusable DbConfig object.

**ConfigSettings**

The DbConfig object to use to configure the database connection properties for this control.

**Events****ProductClick**

```
public event ProductClick(int ProductID)
```

This event fires when the user clicks a Product in the control. This occurs regardless of whether the click is a left or right mouse click, but only if the user clicks a Product.

**ProductID**

This is the ID of the Product clicked by the user.

**ProductDbClick**

```
public event ProductDbClick(int ProductID)
```

This event fires when the user double-clicks a Product in the control. This occurs regardless of whether the double-click is a left or right mouse double-click, but only if the user double-clicks a Product.

**ProductID**

This is the ID of the Product double-clicked by the user.

**ProductGroupClick**

```
public event ProductGroupClick(int ProductGroupID)
```

This event fires when the user clicks a Product Group in the control. This occurs regardless of whether the click is a left or right mouse click, but only if the user clicks a Product Group.

**ProductGroupID**

This is the ID of the Product Group clicked by the user.

**ProductGroupDbClick**

```
public event ProductGroupDbClick(int ProductGroupID)
```

This event fires when the user double-clicks a Product Group in the control. This occurs regardless of whether the double-click is a left or right mouse click, but only if the user double-clicks a Product Group.

#### **ProductGroupID**

This is the ID of the Product Group double-clicked by the user.

#### **ProductGroupSelected**

```
public event ProductGroupSelected(int ProductGroupID)
```

This event fires when the user selects a Product Group. The user can select a Product Group by clicking on it with the mouse or by navigating through the TreeView with keyboard cursor keys.

#### **ProductGroupID**

This is the ID of the Product Group selected by the user.

#### **ProductSelected**

```
public event ProductSelected(int ProductID)
```

This event fires when the user selects a Product. The user can select a Product by clicking on it with the mouse or by navigating through the TreeView with keyboard cursor keys.

#### **ProductID**

This is the ID of the Product selected by the user.

### **Tag Data Control**

The Tag Data control provides programmatic access to the various properties of a Tag. The control has no visual interface and is analogous to an object library rather than a visual component.

The Tag Data control can load any Tag and have access to the majority of its properties. Only Virtual Tags' values are changeable. The Tag Data control offers more advanced Tag processing, while the Tag List Control is designed for providing Tag statuses and making Tag updates in real-time.

### **Properties**

#### **AccessName**

```
public string AccessName { get; }
```

Returns the Name of the Access Name for the loaded Tag.

#### **AccessNameID**

```
public int AccessNameID { get; }
```

Returns the ID of the Access Name for the loaded Tag.

## **DataType**

```
public int DataType { get; }
```

Returns the Data Type of the loaded Tag.

## **DataTypeName**

```
public string DataTypeName { get; }
```

Returns the Data Type Name of the loaded Tag.

## **DefaultValue**

```
public string DefaultValue { get; }
```

Returns the Default Value of the loaded Tag.

## **Description**

```
public string Description { get; }
```

Returns the Description of the loaded Tag.

## **IsVirtualTagType**

```
public bool IsVirtualTagType { get; }
```

Returns whether or not the loaded Tag is a Virtual Tag. This helps verify that Tags are Virtual without performing comparisons to the TagType property.

## **Notes**

```
public string Notes { get; }
```

Returns the Notes for the loaded Tag.

## **Quality**

```
public int Quality { get; }
```

Returns the Quality of the loaded Tag.

## **QualityDescription**

```
public string QualityDescription { get; }
```

Returns the Quality Description of the loaded Tag.

## **Store**

```
public bool Store { get; }
```

Returns whether or not LOGICTrak stores the value of the loaded Tag in the database.

### **TagGroupID**

```
public int TagGroupID { get; }
```

Returns the Tag Group ID of the loaded Tag.

### **TagGroupName**

```
public string TagGroupName { get; }
```

Returns the Tag Group Name of the loaded Tag.

### **TagID**

```
public int TagID { get; }
```

Returns the ID of the loaded Tag.

### **TagName**

```
public string TagName { get; }
```

Returns the Name of the loaded Tag.

### **TagType**

```
public int TagType { get; }
```

Returns the Type of the loaded Tag.

### **TagTypeName**

```
public string TagTypeName { get; }
```

Returns the Type Name for the loaded Tag.

### **UpdateDateTime**

```
public string UpdateDateTime { get; }
```

Returns the last date and time the loaded Tag was updated.

### **Value**

```
public string Value { get; set; }
```

Gets or sets the Value of the loaded Tag. The Tag must be loaded and be a Virtual Tag. In addition, the new value must match the appropriate Data Type. The control does not save the new Value of the Virtual Tag automatically; once successfully set, you must call the Save method to update the value in the database.

### **ValueDiscrete**

```
public bool ValueDiscrete { get; }
```

Returns the Value of the loaded Tag, cast as a Discrete (i.e. Boolean). The Data Type of the Tag must be Discrete in order to ensure a valid result.

### **ValueFloat**

```
public double ValueFloat { get; }
```

Returns the Value of the loaded Tag, cast as a Float. The Data Type of the Tag must be Float in order to ensure a valid result.

### **ValueInteger**

```
public int ValueInteger { get; }
```

Returns the Value of the loaded Tag, cast as an Integer. The DataType of the Tag must be Integer in order to ensure a valid result.

### **ValueString**

```
public string ValueString { get; }
```

Returns the Value of the loaded Tag, cast as a String. The Data Type of the Tag must be String in order to ensure a valid result.

## **Methods**

### **LoadByID**

```
public bool LoadByID([int TagID])
```

This method loads the control with all of the necessary data.

#### **TagID**

This is the ID of the Tag to load.

### **LoadByName**

```
public bool LoadByName([string TagName])
```

This method loads the control with all of the necessary data.

#### **TagName**

This is the Name of the Tag to load.

### **RecordTagHistory**

```
public bool RecordTagHistory([int TagHistoryDefinitionID], [string TagValue],  
[string RecordedDateTime])
```

This method inserts a new Tag History entry as specified by the parameters. Note that it is not necessary for the Tag Data control to be loaded with a specific Tag for this method to work. The return value is False if the insert fails.



**TagHistoryDefinitionID**

This ID specifies the Tag History Definition for which the new entry will be inserted. If this value is passed as -1, then the first Tag History Definition (based on Display Order) for the Tag loaded by the Tag Data control will be used.

**TagValue**

This is the value that will be recorded for the entry.

**RecordedDateTime**

This is the date and time that will be used as the timestamp for the entry. If this value is passed as an empty string (""), then the current TrakSYS date/time will be used as the timestamp (based on the time at the database server).

**Save**

```
public bool Save()
```

This method saves the Value of the loaded Virtual Tag to the database.

**SetPropertiesFromDbConfig**

```
public bool SetPropertiesFromDbConfig(DbConfig ConfigSettings)
```

This method will configure all of the control's database connection properties in one step via a reusable DbConfig object.

**ConfigSettings**

The DbConfig object to use to configure the database connection properties for this control.

## Tag List Control

The Tag List control provides constant programmatic access to the values and states of multiple Tags. The control primarily facilitates constantly updating a view of specific Tags, saving Virtual tags, and displaying Tags in a list.

### Properties

**BackColor**

```
public int BackColor { get; set; }
```

Gets or sets the background color for display text and graphics in the control.

**BadQualityForeColor**

```
public int BadQualityForeColor { get; set; }
```

Gets or Sets the color of the text the control uses to display Bad Quality Tags. This color only applies to Tags with Bad Quality. The control determines if the Tag is Bad using the Quality Mask of the Tag's Access Name, or 192 if the Access Name does not have a Quality Mask.

### **BadQualityIsBold**

```
public bool BadQualityIsBold { get; set; }
```

Gets or sets whether or not to display Tags in the control using bold text. This setting only applies to Tags with Bad Quality. The control determines if the Tag is Bad using the Quality Mask of the Tag's Access Name, or 192 if the Access Name does not have a Quality Mask.

### **ColumnIDWidth**

```
public float ColumnIDWidth { get; set; }
```

Gets or sets the width of the ID Column. To hide this column, set the width to 0.

### **ColumnNameWidth**

```
public float ColumnNameWidth { get; set; }
```

Gets or sets the width of the Name Column. To hide this column, set the width to 0.

### **ColumnQualityWidth**

```
public float ColumnQualityWidth { get; set; }
```

Gets or sets the width of the Quality Column. To hide this column, set the width to 0.

### **ColumnUpdateDateTimeWidth**

```
public float ColumnUpdateDateTimeWidth { get; set; }
```

Gets or sets the width of the Update Date/Time Column. To hide this column, set the width to 0.

### **ColumnValueWidth**

```
public float ColumnValueWidth { get; set; }
```

Gets or sets the width of the Value Column. To hide this column, set the width to 0.

### **Font**

```
public Font Font { get; set; }
```

Gets or sets the font EVENTTrak uses to display text on the control. This is a standard object which provides access to every available font property. If the ActiveX container does not support objects, you can use the other Font properties to adjust the control's font.

### **FontBold**

```
public bool FontBold { get; set; }
```

Gets or sets whether or not the control's text will appear bold.

## FontCharset

```
public int FontCharset { get; set; }
```

Gets or sets the character set the control uses to display text. This property primarily helps support various language settings.

## FontName

```
public string FontName { get; set; }
```

Gets or sets the name of the font the control uses to display text.

## FontSize

```
public float FontSize { get; set; }
```

Gets or sets the font size the control uses to display text.

## GetTagLastCallErrored

```
public bool GetTagLastCallErrored { get; }
```

Returns whether or not the last call to one of the GetTag\* methods resulted in an error. This determines whether any of the methods begin with GetTag\* for this control failed while trying to obtain data. Since the control is designed to run in an HMI container and raising Error in such environments is not recommended, this property is available to perform more advanced error troubleshooting.

## GoodQualityForeColor

```
public int GoodQualityForeColor { get; set; }
```

Gets or sets the color of the text the control uses to display Good Quality Tags. The control determines if the Tag is Good using the Quality Mask of the Tag's Access Name, or 192 if the Access Name does not have a Quality Mask.

## GoodQualityIsBold

```
public bool GoodQualityIsBold { get; set; }
```

Gets or sets whether or not the control uses bold text to display Good Quality Tags. The control determines if the Tag is Good using the Quality Mask of the Tag's Access Name, or 192 if the Access Name does not have a Quality Mask.

## GridLines

```
public bool GridLines { get; set; }
```

Gets or sets whether or not grid lines appear between rows and columns.

## HideColumnHeaders

```
public bool HideColumnHeaders { get; set; }
```

Gets or sets whether or not the control's column headers are hidden or visible.

### **IsDbConnected**

```
public bool IsDbConnected { get; }
```

Returns whether or not the control has successfully connected to the SQL Server database.

### **IsPolling**

```
public bool IsPolling { get; }
```

Returns whether or not the control is actively polling the SQL Server database. If this value is false, then the control is not automatically refreshing the data in the control. To change the value of this property, call the StartPolling or StopPolling methods.

### **PollIntervalSeconds**

```
public int PollIntervalSeconds { get; set; }
```

Gets or sets the time (in seconds) between automatic calls to the PollData method when the control is actively polling.

### **TagCount**

```
public int TagCount { get; }
```

Returns the total number of Tags that are in the control.

## **Methods**

### **AddTagByID**

```
public void AddTagByID([int TagID])
```

This method adds the specified Tag to the internal list of Tags the control is currently monitoring.

#### **TagID**

The ID of the Tag to add to the monitoring list.

### **AddTagByName**

```
public void AddTagByName([string TagName])
```

This method adds the selected Tag to the internal list of Tags the control is currently monitoring.

#### **TagName**

The Name of the Tag to add to the monitoring list.

## AddTagsByTagGroupID

```
public void AddTagsByTagGroupID([int TagGroupID])
```

This method adds all of the tags for the specified Tag Group to the internal list of Tags the control is currently monitoring.

### TagGroupID

The ID of the Tag Group to add to the monitoring list.

## Clear

```
public void Clear()
```

This method Clears all of the Tags and their subscriptions from the control.

## GetDbDateTime

```
public string GetDbDateTime([string TimeFormat=""])
```

This returns the value of the current Date/Time of the SQL Server host. This clock is the basis for all TrakSYS operations.

### TimeFormat

This parameter is a string specifying how to format the Date/Time as a string. The format uses standard Microsoft Visual Basic formatting.

## GetSelectedTagID

```
public int GetSelectedTagID()
```

This returns the ID of the selected Tag in the control. If there are no Tags in the control, the result will be -1.

## GetTagDataTypeByID

```
public int GetTagDataTypeByID([int TagID], [int ErrorTagValue])
```

This returns the DataType of the specified Tag. If for some reason the method fails, it will return the value of the ErrorTagValue instead, and the value of the GetTagLastCalledErrored property will be True.

### TagID

The ID of the desired Tag.

### ErrorTagValue

The value to return if an error occurs. The default value is 0.

## GetTagDataTypeByName

```
public int GetTagDataTypeByName([string TagName], [int ErrorTagValue])
```

This returns the DataType of the specified Tag. If for some reason the method fails, it will return the value of the ErrorTagValue instead, and the value of the GetTagLastCalledErrored property will be True.

**TagName**

The Name of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is 0.

**GetTagDataTypeNameByID**

```
public string GetTagDataTypeNameByID([int TagID], [string ErrorTagValue])
```

This returns the DataTypeName of the specified Tag. If for some reason the method fails, it will return the value of the ErrorTagValue instead, and the value of the GetTagLastCalledErrored property will be True.

**TagID**

The ID of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is "ERROR".

**GetTagDataTypeNameByName**

```
public string GetTagDataTypeNameByName(  
    [string TagName],  
    [string ErrorTagValue]  
)
```

This returns the DataTypeName of the specified Tag. If for some reason the method fails, it will return the value of the ErrorTagValue instead, and the value of the GetTagLastCalledErrored property will be True.

**TagName**

The Name of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is "ERROR".

**GetTagIDByIndex**

```
public int GetTagIDByIndex([int Index])
```

This returns the Tag ID according to the specified index. Although the parameter is optional, it is only optional for compatibility with substandard Active X containers. If the Index parameter is omitted, this method will return a value of -1.

**Index**

This is the 0-based index of the appropriate item in the control.

## GetTagIDByName

```
public int GetTagIDByName([string TagName])
```

This method retrieves the ID of a monitored Tag by its Name.

### TagName

The Name of a monitored Tag.

## GetTagNameByID

```
public string GetTagNameByID([int TagID])
```

This method retrieves the Name of a monitored Tag by its ID.

### TagID

The ID of a monitored Tag.

## GetTagQualityByID

```
public int GetTagQualityByID([int TagID], [int ErrorTagValue])
```

This returns the Quality of the specified Tag. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True.

### TagID

The ID of the Tag desired Tag.

### ErrorTagValue

The value to return if an error occurs. The default value is 0.

## GetTagQualityByName

```
public int GetTagQualityByName([string TagName], [int ErrorTagValue])
```

This method returns the Quality of the specified Tag. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True.

### TagName

The Name of the desired Tag.

### ErrorTagValue

The value to return if an error occurs. The default value is 0.

## GetTagUpdateDateTimeByID

```
public string GetTagUpdateDateTimeID(  
    [int TagID],  
    [string ErrorTagValue],  
    [string TimeFormat]  
)
```

This returns the Update Date/Time of the specified Tag. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True.

**TagID**

The ID of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is "ERROR".

**TimeFormat**

This parameter is a string specifying how to format the Date/Time as a string. The format uses standard Microsoft Visual Basic formatting.

**GetTagUpdateDateTimeByName**

```
public string GetTagUpdateDateTimeName(  
    [string TagName],  
    [string ErrorTagValue],  
    [string TimeFormat]  
)
```

This returns the Update Date/Time of the specified Tag. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True.

**TagName**

The Name of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is "ERROR".

**TimeFormat**

This parameter is a string specifying how to format the Date/Time as a string. The format uses standard Microsoft Visual Basic formatting.

**GetTagValueByID**

```
public string GetTagValueByID([int TagID], [string ErrorTagValue])
```

This returns the Value of the specified Tag. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True.

**TagID**

The ID of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is "ERROR".

**GetTagValueByName**

```
public string GetTagValueByName([string TagName], [string ErrorTagValue])
```



This returns the Value of the specified Tag. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True.

**TagName**

The Name of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is "ERROR".

**GetTagValueDiscreteByID**

```
public bool GetTagValueDiscreteByID([int TagID], [bool ErrorTagValue])
```

This returns the Value of the specified Tag, cast to a Discrete. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True. The method will fail if the Data Type of the Tag is not Discrete.

**TagID**

The ID of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is False.

**GetTagValueDiscreteByName**

```
public bool GetTagValueDiscreteByName(  
    [string TagName],  
    [bool ErrorTagValue]  
)
```

This returns the Value of the specified Tag, cast to a Discrete. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True. The method will fail if the Data Type of the Tag is not Discrete.

**TagName**

The Name of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is False.

**GetTagValueFloatByID**

```
public double GetTagValueFloatByID([int TagID], [double ErrorTagValue])
```

This returns the Value of the specified Tag, cast to a Float. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True. The method will fail if the Data Type of the Tag is not Float.

**TagID**

The ID of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is -1.

**GetTagValueFloatByName**

```
public double GetTagValueFloatByName(  
    [string TagName],  
    [double ErrorTagValue]  
)
```

This returns the Value of the specified Tag, cast to a Float. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True. The method will fail if the Data Type of the Tag is not Float.

**TagName**

The Name of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is -1.

**GetTagValueIntegerByID**

```
public int GetTagValueIntegerByID([int TagID], [int ErrorTagValue])
```

This returns the Value of the specified Tag, cast to an Integer. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True. The method will fail if the Data Type of the Tag is not Integer.

**TagID**

The ID of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is -1.

**GetTagValueIntegerByName**

```
public int GetTagValueIntegerByName([string TagName], [int ErrorTagValue])
```

This returns the Value of the specified Tag, cast to an Integer. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True. The method will fail if the Data Type of the Tag is not Integer.

**TagName**

The Name of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is -1.

**GetTagValueStringByID**

```
public string GetTagValueStringByID([int TagID], [string ErrorTagValue])
```

This returns the Value of the specified Tag, cast to a String. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True. The method will fail if the Data Type of the Tag is not String.

**TagID**

The ID of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is "ERROR".

**GetTagValueStringByName**

```
public string GetTagValueStringByName(  
    [string TagName],  
    [string ErrorTagValue]  
)
```

This returns the Value of the specified Tag, cast to a String. If for some reason the method fails, it will return the value of ErrorTagValue instead, and the value of the GetTagLastCallErrored property will be True. The method will fail if the Data Type of the Tag is not String.

**TagName**

The Name of the desired Tag.

**ErrorTagValue**

The value to return if an error occurs. The default value is "ERROR".

**PollData**

```
public void PollData()
```

This method is the main process of the Tag List control. This retrieves all of the data for each subscribed Tag. The control automatically calls this method if it is actively polling (IsPolling is True). In addition, this method raises the 2 events, BeforePollData and AfterPollData, to allow the ActiveX container to perform any necessary pre-poll or post-poll processing.

**RemoveTagByID**

```
public void RemoveTagByID([int TagID])
```

This method removes the specified Tag from the internal list of Tags the control is currently monitoring. This method only removes Tags added by the AddTagByID method.

**TagID**

The ID of the Tag to remove from the monitoring list.

**RemoveTagByName**

```
public void RemoveTagByID([string TagName])
```

This method removes the specified Tag from the internal list of Tags the control is currently monitoring. This method only removes Tags added by the AddTagByName method.

**TagName**

The Name of the Tag to remove from the monitoring list.

**RemoveTagsByTagGroupID**

```
public void RemoveTagsByTagGroupID([int TagGroupID])
```

This method removes all Tags belonging to the specified Tag Group from the internal list of Tags the control is currently monitoring. This method only removes Tags added by the AddTagsByTagGroupID method.

**TagGroupID**

The ID of the Tag Group to remove from the monitoring list.

**SaveTagValueByID**

```
public bool SaveTagValueByID([int TagID], [string TagValue])
```

This method attempts to update the Value of the specified Tag. This method will only be successful if the Tag is Virtual, the TagValue is valid for the Tag's Data Type, and the Tag is currently in the list of Tags that the control is monitoring.

**TagID**

The ID of the Tag to change.

**TagValue**

The desired new value of the Tag.

**SaveTagValueByName**

```
public bool SaveTagValueByID([string TagName], [string TagValue])
```

This method attempts to update the Value of the specified Tag. This method will only be successful if the Tag is Virtual, the TagValue is valid for the Tag's Data Type, and the Tag is currently in the list of Tags that the control is monitoring.

**TagName**

The Name of the Tag to change.

**TagValue**

The desired new value of the Tag.

**StartPolling**

```
public void StartPolling()
```

This method attempts to begin actively polling the database. This uses the PollIntervalSeconds property to determine the time between automatic calls to the PollData method.

## StopPolling

```
public void StopPolling()
```

This method will cease any ongoing, automatic polling.

## SetPropertiesFromDbConfig

```
public bool SetPropertiesFromDbConfig(DbConfig ConfigSettings)
```

This method will configure all of the control's database connection properties in one step via a reusable DbConfig object.

## ConfigSettings

The DbConfig object to use to configure the database connection properties for this control.

## Events

### AfterPollData

```
public event AfterPollData()
```

This event fires after the PollData method completes, regardless of whether the call to PollData occurs automatically or manually. If you want to capture Tag changes, use the TagChanged event instead.

### BeforePollData

```
public event BeforePollData()
```

This event fires before the PollData method completes all of its processing, regardless of whether the call to PollData occurs automatically or manually.

### DbDisconnected

```
public event DbDisconnected()
```

This event fires if the control encounters an error attempting to establish a connection to the SQL Server database. It allows the programmer to react to database connectivity errors without constantly checking the IsDbConnected property value.

### DbReconnected

```
public event DbReconnected()
```

This event fires if the control encounters an error attempting to connect to the SQL Server database and then successfully reconnects. This event will fire if the control successfully reconnects during the current its lifespan. If the container application repeatedly loads and unloads the control, this event may never fire.

## TagClick

```
public event TagClick(int TagID)
```

This event fires when the user clicks a Tag in the control. This occurs regardless of whether the click is a left or right mouse click, but only if the user clicks a Tag.

### TagID

This is the ID of the Tag clicked by the user.

## TagDoubleClick

```
public event TagDoubleClick(int TagID)
```

This event fires when the user double-clicks a Tag in the control. This occurs regardless of whether the double-click is a left or right mouse double-click, but only if the user double-clicks a Tag.

### TagID

This is the ID of the Tag double-clicked by the user.

## TagSelected

```
public event TagSelected(int TagID)
```

This event fires when the user selects a Tag. The user can select a Tag by clicking on it with the mouse or by navigating through the list with the keyboard cursor keys.

### TagID

This is the ID of the Tag selected by the user.

## TagChanged

```
public event TagChanged(  
    int TagID,  
    string Value,  
    string LastValue,  
    int Quality,  
    int LastQuality  
)
```

This event fires when a monitored Tag's value or quality changes.

### TagID

The ID of the Tag that has changed.

### Value

The new Value of the Tag.

### LastValue

The previous Value of the Tag. This may be identical to the current Value if only the Quality of the Tag has changed.

### Quality

The new Quality of the Tag.

### **LastQuality**

The previous Quality of the Tag. This may be identical to the current Quality if only the Value of the Tag has changed.

## **Touchboard Control**

The Touchboard Control allows users to input text using a software-based keyboard. This control launches a modal window with a keyboard, allowing the user to input text. This control does not share any of the Common Properties with the other EVENTTrak Controls.

### **Methods**

#### **GetCurrentLCID**

```
public int GetCurrentLCID()
```

This returns the current Locale Identifier (LCID) that the machine is currently using.

#### **GetInput**

```
public string GetInput([int LCID], [string InitialText])
```

This will launch a modal window representing a software keyboard. It includes a textbox which users can edit using the onscreen virtual keys. The default value of the textbox is set to the InitialText parameter. If the user presses the confirmation button, the GetInput method will return the modified text; otherwise, if Cancel is pressed, this method will return the value of the InitialText parameter.

If an unsupported LCID is requested, the control will default to the United States locale (1033).

#### **LCID**

The Locale Identifier that selects a localized keyboard layout for a Language – Country/Region.

#### **InitialText**

This specifies the default text for the user to modify, as well as what the method will return if the user presses Cancel.

## **Sample Sub-Group Data Control**

The Sample Sub-Group Data control provides programmatic access to writing sample data for SPC analysis. The data in this control represents a single sub-group of sample data for a specified Sample Definition from the configuration.

### **Properties**

#### **ID**

```
public int ID { get; }
```

Returns the ID for the Sample Sub-Group entry. This ID is -1 until the Save() method is called at which point a new ID is generated. It will also be populated after the LoadByID() method is called.

### Identifier

```
public string Identifier { get; set; }
```

Gets or sets the unique identifier that will be recorded with the Sample Sub-Group data. This identifier must be unique for all Sub-Groups within a specified Sample Definition.

### Notes

```
public string Notes { get; set; }
```

Gets or sets the notes regarding the Sample Sub-Group entry. This property is not required and can have any value.

### ProcessMean

```
public double ProcessMean { get; set; }
```

Gets or sets the Process Mean value that will be stored with the recorded Sample Sub-Group. This property can be set manually or automatically by calling the *SetLimits* method of this control.

### ProcessLCL

```
public double ProcessLCL { get; set; }
```

Gets or sets the Process LCL value that will be stored with the recorded Sample Sub-Group. This property can be set manually or automatically by calling the *SetLimits* method of this control.

### ProcessUCL

```
public double ProcessUCL { get; set; }
```

Gets or sets the Process UCL value that will be stored with the recorded Sample Sub-Group. This property can be set manually or automatically by calling the *SetLimits* method of this control.

### ProcessLSL

```
public double ProcessLSL { get; set; }
```

Gets or sets the Process LSL value that will be stored with the recorded Sample Sub-Group. This property can be set manually or automatically by calling the *SetLimits* method of this control.

### ProcessUSL

```
public double ProcessUSL { get; set; }
```



Gets or sets the Process USL value that will be stored with the recorded Sample Sub-Group. This property can be set manually or automatically by calling the *SetLimits* method of this control.

### **SampleCount**

```
public int SampleCount { get; set; }
```

Gets or sets the total number of samples being examined. This property is used when the Sample Definition is in Percentage mode and Sample Values are only being recorded for samples that fail inspection. In this case, the SampleCount might be set to 10 but only 3 samples failed inspection so the SampleValue property would return 3.

### **SampleDateTime**

```
public string SampleDateTime { get; set; }
```

Gets or sets the date and time stamp that will be recorded with the Sample Sub-Group data.

### **SampleDefinitionID**

```
public int SampleDefinitionID { get; set; }
```

Gets or sets the ID corresponding to the parent Sample Definition that the Sub-Group data will be recorded for. This property must be set before the Save() method is called for data to be recorded properly.

### **SampleValues**

```
public int SampleValues { get; }
```

Returns the number of Sample Value entries currently loaded into the control (based on the number of times the AddSample () method is called). This value may be different from the SampleCount property (see the SampleCount property description for more details).

### **User**

```
public string User { get; set; }
```

Gets or sets the user name for the person making the Sample Sub-Group data entry. This property is not required and can have any value.

### **JobID**

```
public int JobID { get; set; }
```

Gets or sets the ID of the Job for the Sample Sub-Group entry.

### **BatchID**

```
public int BatchID { get; set; }
```

Gets or sets the ID of the Batch for the Sample Sub-Group entry.

## Methods

### AddSample

```
public bool AddSample([double Value], [int CategoryID])
```

This method adds a new Sample value to the Sub-Group with the specified Sample Category. Sample values are not saved from the database until the Save() method is called. The return value is False if the load fails.

#### Value

This is the value for the Sample that is being added.

#### CategoryID

This is the ID for the Sample Category that will be applied to the Sample being added. The default value is -1 indicating that no Category should be applied.

### Clear

```
public bool Clear()
```

This method clears all of the properties of the control and returns it to its initial empty state.

### Delete

```
public bool Delete()
```

This method removes the Sample Sub-Group and all Samples for the Sample Sub-Group ID that is set to the control (the ID property of this control). This method returns False if delete fails.

### DeleteWithAudit

```
public bool DeleteWithAudit([string User01], [string User02], [string Notes])
```

This method is the same as the Delete() method but inserts the audit information passed in and associates it with the update. This method should only be used if AUDITTrak is licensed and enabled. The return value is False if the delete fails.

#### User01 and User02

These values are the primary and secondary signature users for the audit action.

#### Notes

This is any notes that are to be associated with the audit action.

### IsIdentifierUnique

```
public bool IsIdentifierUnique()
```

This method returns True if the Identifier property that is set to the control passes a uniqueness validation check for all other Sub-Groups for the currently set Sample Definition. This method returns False if the Identifier is not unique.

### **GetSampleCategory**

```
public int GetSampleCategory([int Index])
```

This method returns the Sample Category ID for the specified Sample.

#### **Index**

This is the position of the Sample that is to be accessed. The collection is 1 based (starts at 1) and holds the number of Samples specified by the SampleValues property.

### **GetSampleValue**

```
public double GetSampleValue([int Index])
```

This method returns the Sample value for the specified Sample.

#### **Index**

This is the position of the Sample that is to be accessed. The collection is 1 based (starts at 1) and holds the number of Samples specified by the SampleValues property.

### **LoadByID**

```
public bool LoadByID([int SampleSubGroupID])
```

The method loads the control with the data from the specified Sample Sub-Group. The return value is False if the load fails.

### **SampleSubGroupID**

This is the ID of the SampleSubGroup that will be loaded.

### **RemoveSample**

```
public bool RemoveSample([int Index])
```

This method removes Sample from the control. Sample values are not removed from the database until the Save() method is called. Also, the return value of the SampleValues property is not decremented until the Save() method is called. The return value is False if the remove fails.

#### **Index**

This is the position of the Sample value that is to be removed. The collection is 1 based (starts at 1) and holds the number of Samples specified by the SampleValues property.

## SetLimits

```
public bool SetLimits()
```

This method populates the limit properties of this control (Process Mean, LCL, UCL, LSL, USL) by retrieving the constant or Tag values specified in the Sample Definition. This method should be called immediately before the Save method. If the limits are set manually, then this method does not need to be called.

## Save

```
public bool Save()
```

This method saves all of the set property and Sample value data to the database. If the Sample Sub-Group ID (the ID property of this control) is -1 then a new Sample Sub-Group is created and new Sample records are added. If the Sample Sub-Group ID is not -1, then the specified Sub-Group and its Samples are updated accordingly. This method returns False if save fails.

## SaveWithAudit

```
public bool SaveWithAudit([string User01], [string User02], [string Notes])
```

This method is the same as the Save() method but inserts the audit information passed in and associates it with the update. This method should only be used if AUDITTrak is licensed and enabled. The return value is False if the save fails.

### User01 and User02

These values are the primary and secondary signature users for the audit action.

### Notes

This is any notes that are to be associated with the audit action.

## SetSampleCategory

```
public bool SetSampleCategory([int Index], [int CategoryID])
```

This method sets a Sample Category to the specified Sample. Samples are not saved to the database until the Save() method is called. The return value is False if the set fails.

### Index

This is the position of the Sample that is to be accessed. The collection is 1 based (starts at 1) and holds the number of Samples specified by the SampleValues property.

### CategoryID

This is the ID for the Sample Category that will set. To clear the Category set this value to -1.

## SetSampleValue

```
public bool SetSampleValue([int Index], [double Value])
```

This method sets a value to the specified Sample. Samples are not saved to the database until the Save() method is called. The return value is False if the set fails.

### Index

This is the position of the Sample that is to be accessed. The collection is 1 based (starts at 1) and holds the number of Samples specified by the SampleValues property.

### Value

This is the value that is to be set to the Sample.

## ValidateAuditorLogin

```
public bool ValidateAuditorLogin ([string Login], [string Password])
```

This method validates a login and password based on the Authentication Type set in the TrakSYS settings. The return value is True if the login and password is valid and the user is configured as an Auditor.

### Login

This is the login as specified in a TrakSYS user record. For Windows Users, the login should include the domain in the format DOMAIN\login.

### Password

This is the TrakSYS or Windows password for the Login passed in. The password is case sensitive.

## ValidateWindowsLogin

```
public bool ValidateWindowsLogin ([string Login], [string Password])
```

This method validates a Windows login and password. The return value is True if the login exists in the specified domain and the password is correct.

### Login

This is a Windows login including the domain in the format DOMAIN\login.

### Password

This is the Windows password for the Login passed in. The password is case sensitive.

## Time Span Format

Time Span formatting allows the programmer more flexibility in displaying durations (i.e. Time Spans). There are several properties and values in which the controls return results in seconds. This custom format allows programmers to display the values in a variety of ways.

The Time Span custom format string has several reserved, case-sensitive characters that constitute its formatting. If a control receives a format string without these characters, it will treat them as string literals, displaying them as provided.

If you specify an empty format string, EVENTTrak will treat it as the default format "hh:nn:ss".

All of the following examples are based on the value of 3,811 seconds (which is 1 hour, 3 minutes and 31 seconds).

hh	Hours padded with zero. Example: "01"
nn	Minutes padded with zero. Example: "03"
ss	Seconds padded with zero. Example: "31"
h	Hours with no padding. Example: "1"
n	Minutes with no padding. Example: "3"
s	Seconds with no padding. Example: "31"
H2	Total hours with up to 2 precision. Example: "1.06"
H1	Total hours with up to 1 precision. Example: "1.1"
H0	Total hours with no precision. Example: "1"
H	same as H0
N2	Total minutes with up to 2 precision. Example: "63.52"
N1	Total minutes with up to 1 precision. Example: "63.5"
N0	Total minutes with no precision. Example: "63"
N	same as N0
S	Seconds. Example: "3811"

## Localization

The EVENTTrak controls support localization via special RESX files. These files are installed alongside the EVENTTrak controls into the TrakSYS Program Files folder. They are standardized XML files containing translations for all visible strings in the controls.

The EVENTTrak controls have their own set of RESX files. The table below illustrates the naming convention for these files, and the order in which the controls search for an applicable localized RESX file.

<b>EVENTTrak Control Set Translations</b>	
EMX.{LCID}.resx	Translation file for a particular LCID, i.e. 1033 for U.S. English.
EMX.{primary language ID}.resx	Translation file for a primary language, i.e. 9 for English.
EMX.resx	Fallback translation file, which is U.S. English by default.

## **Sample Applications**

### **Sample C# EVENTTrak Application**

TrakSYS includes a comprehensive sample EVENTTrak application and its C# source code for the .NET framework. This sample demonstrates using the new set of controls to display, categorize, and Events, as well as how to trigger Manual Events and set the current Job and Batch. The sample application is based on the Sample configuration that can be optionally installed when creating a new EDB database. You will need Visual Studio 2008 in order to open and edit the sample application.

## **WEBTrak Knowledge Management Portal**

WEBTrak Knowledge Management Portal is a comprehensive reporting portal with its own security framework and diverse set of reports. You can customize the prepackaged reports and tailor them to specific users' needs or simply use the standard ones to immediately begin analyzing your data.

Some reports may appear in pop-up windows, so you may need to configure your browser to allow pop-ups from the WEBTrak site. You can do this in Internet Explorer by clicking Tools | Pop-up Blocker | Pop-up Blocker Settings and adding the WEBTrak server to the list of Allowed sites.

## **Security**

WEBTrak relies on Windows Integrated Security or TrakSYS logins and passwords to control access to the site itself, but uses its own permissions scheme to control what users can see and access within the site. By default, WEBTrak grants administrative rights to everyone, so it is important that a qualified administrator creates separate administrator and user accounts to secure the site after installation.

Access to WEBTrak involves two distinct forms of security. The first, authentication, is what controls access to the site itself. Unauthenticated users cannot reach any part of the site and will receive an access denied error page (Windows Security) or a login screen (TrakSYS Security) when they attempt to connect.

The second form of security is authorization, which determines what rights users have within the WEBTrak site. WEBTrak uses the authentication credentials to authorize access to administrative tools and Report Groups. Administrators define users according to TrakSYS Users, Windows Users, Windows Groups, computer names, IP addresses, or private unique identifiers, and then assign those Report Users access rights to individual Report Groups. Refer to the section on Report Users for more information on creating and managing Report Users.

WEBTrak must be configured for either Windows or TrakSYS authentication (not both).

### **Configuring WEBTrak for Integration with Windows Security**

In order to use Windows Users and Groups with WEBTrak security, the site must be configured to use Windows authentication.

The first step is done via the Internet Information Services Manager. Locate the WEBTrak application in the Tree View and select Properties from the right-click

menu. From the Directory Security tab, click the Edit button in the Anonymous Access and Authentication Control section. Uncheck the Anonymous Access checkbox at the top of the Authentication Methods dialog and make sure the Integrated Windows Authentication checkbox at the bottom is checked. Once this is done, WEBTrak can evaluate web client rights based on the Windows User or Group being used to access the site. The Windows User for the user can be determined by the User Information page available from the Diagnostics menu.

The second step is to change the authentication type in the web.config file for the WEBTrak site. This file is located in the installation directory (typically C:\Program Files\Parsec\TrakSYS\WEBTrak). The value for the "mode" attribute of the "authentication" node must be changed to "Windows". The change is effective once the file is saved.

## **Configuring WEBTrak for Integration with TrakSYS Security**

In order to use logins and passwords defined within MODELTrak to access the WEBTrak site, it must be configured to use Forms authentication.

For TrakSYS Security, change the authentication type to Forms in the web.config file in the for the WEBTrak site. This file is located in the installation directory (typically C:\Program Files\Parsec\TrakSYS\WEBTrak). The value for the "mode" attribute of the "authentication" node must be changed to "Forms". The change is effective once the file is saved.

## **Administration**

This section of WEBTrak allows Administrators to make site-wide adjustments to reports and to configure security.

## **Reporting Fundamentals**

### **Report Types**

Report Types are the foundation for reports in WEBTrak. A Report Type is a template for an actual report; it defines the underlying report's parameters, filters, and location. You cannot directly create Report Types in WEBTrak, but you can modify their parameters in the Administration section of the site.

Using one of the Report Types as a template, a Report selectively overrides the default parameter values and adjusts which parameters end users will see. Reports simplify the user experience by narrowing the end results in anticipation of users' needs. For example, one Report might offer an event summary for a single production line on the current day, sparing users from having to repeatedly select the same line and date range to obtain the desired information.

The sections below describe the base Report Types. Refer to the Reports section for more information on the available properties and parameters for each Report Type.

### **Common Properties**

This section describes the common properties of every Report Type. Use the Report Type Designer (by selecting Report Types from the Administration menu) to view and edit these properties.



<b>General</b>	
Report Type Group	The parent Report Type Group, which acts as a folder and organizes similar or related reports. Change this value to move this Report Type to another Report Type Group.
Name	Display name of this Report Type as it will appear in WEBTrak.
Description	Description of this Report Type.
Key	Unique string identifier for this Report Type.
Version	Version of this Report Type.
<b>Default Report Values</b>	These properties are default settings for every new Report you create based on the selected Report Type. Modifying these properties will not affect any existing Reports.
Name	Default name for Reports based on this Report Type.
Description	Default description for Reports based on this Report Type.
Url	Default URL for Reports based on this Report Type. This is typically a Reporting Services URL.
Target	Default Target setting for Reports based on this Report Type. This setting determines how Reports will open in the browser. "Same Window" means Reports will open inside WEBTrak. "New Window (Single)" means Reports will open in a new browser window, but will reuse the same new window for each subsequent call to the same Report. "New Window (Multiple)" means each call to the Report will open a new window.
Window	Default parameters to supply to the Report browser window, specifying settings for location, size, appearance, and behavior.
Auto Open	Default Auto-Open setting for Reports, which causes them to automatically open without prompting users for additional parameters or filters.
Open Caption	Label that will appear on the button used to launch Reports based on this Report Type. This typically uses a variable, such as openCaptionOpen, which is typically "Open".
Icon	Default display icon for Reports based on this Report Type.

## Dashboard Reports


The Dashboard Report Types allow for the custom arrangement of smaller report components called Web Parts.

### Dashboard

This Report Type is the standard pallet for creating custom dashboard reports. The Design properties of this report allow for different Zone layouts to be specified.

### Event Reports

These reports display Events recorded by LOGICTrak, organized by Areas, Systems, and other Configuration Entities. Detail views within these reports, accessible via the

 detail icon, display the corresponding Event Definition, Sub-System, operator provided data, and all other information related to an Event.

These reports are an excellent source of information for identifying problematic or under-performing equipment. For OEE and other productivity measurements, refer to the KPI Reports.

### **Area Event Summary (Date)**

This Report Type displays Event data in various formats. The primary filters for this report are Area and Start and End dates, which limit the data to events overlapping a date range for a particular Area.

This Report Type uses the Start and End Date parameters to select Event data based on the EventDate field of the Event data table. LOGICTrak populates this field based on the Event Date property of the Event Definition that triggered the Event.

### **Area Event Summary (DateTime)**

This Report Type displays Event data in various formats. The primary filters for this report are Area and Start and End dates, which limit the data to events overlapping a date range for a particular Area.

This Report Type uses the Start and End Date parameters to select Event data based on the StartDateTime field of the Event data table. LOGICTrak populates this field based on the date and time when the Event starts.

### **System Event Summary (Date)**

This Report Type displays Event data in various formats. The primary filter for this report is System, which limits the data to the selected System.

This Report Type uses the Start and End Date parameters to select Event data based on the EventDate field of the Event data table. LOGICTrak populates this field based on the Event Date property of the Event Definition that triggered the Event.

### **System Event Summary (DateTime)**

This Report Type displays Event data in various formats. The primary filter for this report is System, which limits the data to the selected System.

This Report Type uses the Start and End Date parameters to select Event data based on the StartDateTime field of the Event data table. LOGICTrak populates this field based on the date and time when the Event starts.

### **Mean Time Between Failure (Date)**

This Report Type displays Mean Time Between Failure Event data in various formats. The primary filter for this report is System, which limits the data to the selected System.

See the MTBF Type property of Event Definitions is used to determine how Event counts and duration are applied to this report.

Mean Time Between Failure (MTBF) is defined as the total duration of the Report (Start Date to End Date) divided by the total Event count. This makes MTBF the average time between the start of one Event and the start of the next.

Mean Time Running Between Failure (MTrBF) is defined as the total duration of the Report (Start Date to End Date) minus the total amount of Event seconds, divided by the total Event count. This makes MTrBF the average time between the end of one Event and the start of the next.

This Report Type uses the Start and End Date parameters to select Event data based on the EventDate field of the Event data table. LOGICTrak populates this field based on the Event Date property of the Event Definition that triggered the Event.

## **KPI Reports**

These reports calculate and display standard productivity KPIs, including OEE, TEEP, and Takt Time. For OEE, there are additional reports for Availability, Performance, and Quality. The Event Reports are useful in conjunction with these reports to investigate root causes of productivity losses.

### **OEE Summary**

This Report Type displays aggregate OEE and related KPI data for the selected date range. It also displays initial production counts (Good, Bad and Total) as well as the counts after adjustment.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### **OEE Detail**

This Report Type displays OEE data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### **Availability Detail**

This Report Type displays Availability data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### **Performance Detail**

This Report Type displays Performance data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### **Quality Detail**

This Report Type displays Quality data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### **TEEP Detail**

This Report Type displays TEEP data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### **Takt Detail**

This Report Type displays Takt data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### **Production Loss**

This Report Type displays production data grouped into the selected period for the selected KPI Calculation. The Report contains three sections.

The first section contains a graph showing the total production in relation to the Production and Net Operation targets. The Production target is the number of units the line could produce in the total report time minus System Not Scheduled and Availability Loss Event time. The Net Operation target is the number of units the line could produce in the total report time minus System Not Scheduled Event time.

The second section contains a graph showing the loss cost for each period group. The report breaks the losses down by Availability, Performance, and Quality losses. The Report input parameters specify the cost per unit for each of these loss types.

The third section contains a tabular representation of the data specified by the Report parameters.

### **OEE Comparison Detail**

This Report Type allows the side-by-side comparison of KPI data across several KPI Calculations. KPI Calculations can be grouped for selection by using KPI Calculation Views in MODELTrak. The Report contains three sections.

The first section contains a trend graph with one line per KPI Calculation plotted across the grouping option selected. The average OEE for all selected KPI Calculations is displayed at the top of the chart.

The second section is a tabular list of the OEE rollup for each KPI Calculation for the selected KPI Calculation View.

The third section contains a tabular list of the OEE breakdown for each group in the selected grouping. The OEE for each KPI Calculation is broken out within each grouping.

## **Administration**

These reports are primarily troubleshooting tools. They may be particularly helpful during initial deployment when configuration, communication, and security problems are most likely to occur.

### **Log Summary**

This Report Type displays application log data recorded by individual TrakSYS components. Entries in this report represent informational events, warnings, or errors during execution. All of the applications' log information is available through this report, with the exception of entries occurring while the database is inaccessible.

### **KPI Counter Detail**

This Report Type displays a trend for a specified KPI Counter for a specified KPI Calculation over a period of time. There are two graphs as well as a tabular view of the counter data. The first chart is a trend of the actual Counter values. The second is a trend of the Counter changes (delta).

A KPI Counter will only record the detail information available in this report when its "Record Detail" property is enabled in MODELTrak.

### **KPI Dump**

This Report Type generates an Excel file with all of the KPI Interval records for a selected KPI Calculation over a specified date range.

### **Data Range**

This Report Type is used to list the different ranges of data that exist in the database. Systems and KPI Calculations are listed and the minimum and maximum dates that data has been recorded are displayed for each.

### **Audit Summary**

This Report Type lists a summary of the audit records that are captured by the AUDITTrak feature for a specified date range and filters.

### **Audit Dump**

This Report Type generates an Excel file with all of the audit records over a specified date range and filters.

### **Journal Summary**


This Report Type lists a summary of the Journal records that are entered using the IMPROVETrak Journal entry forms.

## **Editors**

The Editor Reports allow access to the data editing interfaces for Event and KPI data. The parameters for these Report Types allow customization of the editor forms.


### **Event List**

This Report Type displays Events for the selected System and date range. The includes only the parent System event (does not include an expanding section with the sub-events).

Authorized users can click the  edit link to the left of each row to launch the Event editor dialog, where they can edit all of an Event's recorded attributes or delete the Event entirely.


### **Event List with Sub Events**


This Report Type displays Events and their sub-events for the selected System and date range.

Authorized users can click the  edit link to the left of each row to launch the Event editor dialog, where they can edit all of an Event's recorded attributes or delete the Event entirely.

### **KPI Interval List**

This Report Type displays KPI Intervals for the selected KPI Calculation and date range.

Authorized users can click the  edit link to the left of each row can to launch the KPI Interval editor dialog, where then can edit all of a KPI Interval's recorded attributes.

The  adjustment link to the left of each row launches the New KPI Interval Adjustment dialog, allowing users to make counter adjustments to the selected Interval.

### **New Event**

This Report Type launches a New Event data entry dialog. The parameters of the Report Type can specify default values for some of the Event attributes. For instance, the Report Type may pre-select the System and Event Definition so that clicking on the New Event node in the WEBTrak tree view will show an entry form with those attributes already set.

### **New KPI Interval Adjustment**

This Report Type launches a New KPI Interval Adjustment entry dialog for making counter adjustments to a single Interval. The parameters of the Report Type can specify default values for some of the adjustment attributes. For instance, the Report Type may pre-select the KPI Calculation and Adjusted Counter so that clicking on the New KPI Interval Adjustment node in the WEBTrak tree view will show an entry form with those attributes already set.

### **New KPI Range Adjustment**

This Report Type launches a New KPI Range Adjustment entry dialog for making counter adjustments to a range of Intervals. The parameters of the Report Type can specify default values for some of the adjustment attributes. For instance, the Report Type can pre-select the KPI Calculation, Adjusted Counter, and Start/End Date Range so that clicking on the New KPI Range Adjustment node in the WEBTrak tree view will show an entry form with those attributes already set.

The entry form generates a table with each of the Intervals that fall within the specified range. After entering a Calculation Units adjustment at the top of the screen, clicking the Recalculate button will evenly distribute the adjustment across the selected Intervals according to the Production Time of each Interval. You can

also manually modify each Interval's adjustment if necessary. Refer to the section on KPI Units for more information on Calculation Units.

### **New Journal Entry**

This Report Type launches an IMPROVETrak Journal data entry dialog. The parameters of the Report Type can specify default values for some of the Journal entry attributes. For instance, the Report Type may pre-select the System and so that clicking on the New Journal Entry node in the WEBTrak tree view will show an entry form with the System already set.

### **Extensibility**

#### **Link URL**

This Report Type facilitates custom entries in the WEBTrak tree view which link to user defined URLs. There are no parameters for this Report Type for end users to select. Reports of this type typically use the Auto Open setting so that the link opens as soon as a user clicks the corresponding node in the WEBTrak tree view.

#### **URL Link Refresh**

This Report Type provides auto-refresh support to any URL, which allows Reports to periodically regenerate according to a configurable interval without user intervention.

#### **Link Reporting Services**

This Report Type facilitates custom entries in the WEBTrak tree view which link to SQL Server Reporting Services Reports.

This Report Type offers several Reporting Services parameters which control how Reporting Services requests and displays these Reports.

The parameters for this Report Type are not typically available to end users. These Reports typically use the Auto Open setting so that the link opens as soon as a user clicks the corresponding node in the WEBTrak tree view.

#### **Link Report Key**

This Report Type facilitates custom entries in the WEBTrak tree view which link to another Report in the site by Report Key.

The parameters for this Report Type are not typically available to end users. These Reports typically use the Auto Open setting so that the link opens as soon as a user clicks the corresponding node in the WEBTrak tree view.

### **Report Variables**

This Administration page allows you to define Report Variables common to all Report Groups. Use Variable Groups to group together sets of related Report Variables. Event Report Variable belongs to a Variable Group, and will appear in the same hierarchy within the Report Group Variables section.

<b>Report Variable Properties</b>	
Name	Display name of this Report Variable. This name will appear in the Variables section of Report Groups.



Key	Unique string identifying this Report Variable. Use this key to reference Report Variables from Textbox Parameter Types in the Report designer.
Group	Report Variable Group this Report Variable belongs to. The variable will appear in the same Group in the Variables section of every Report Group.

## Report Users

Report Users represent users, groups, and nodes authorized to access various parts of WEBTrak. Report Users only authorize access to Reports and Report Groups within WEBTrak; they do not authenticate users attempting to access the web site. Refer to the section on Security for an overview of security in WEBTrak. A Report User can be one of the following types:

<b>User Types</b>	
TrakSYS User	A login and password that represents an individual user. TrakSYS User login information is created and maintained in MODELTrak. The WEBTrak site must be configured for TrakSYS Security in order for TrakSYS Users to be used for authentication.
TrakSYS Group	A TrakSYS Group name. Use this type to authorize entire groups of TrakSYS Users.
Windows User	<p>A Windows User name. Use this type to force users to provide their Windows login to access WEBTrak. When using these types of users, you should disable anonymous access to the WEBTrak web site using Windows Internet Information Services (IIS) Manager, otherwise users may access the site using the default IUSER_&lt;SERVERNAME&gt; account and they will be indistinguishable from one another.</p> <p>You must grant Windows Users or their Groups access to the WEBTrak web site via the Windows Internet Information Services (IIS) Manager. As the section on Security explains, Report Users only authorize users to access pages within WEBTrak; Report Users do not provide access to the web site itself (authentication).</p>
Windows Group	<p>A Windows Group name. Use this type to authorize entire groups of Windows Users. With Report Users of this type, you can centralize security in Windows Active Directory without explicitly creating Report Users for every end user.</p> <p>The same security constraints for User types apply to Group types as well. Authorizing Groups of Windows users to access various parts of WEBTrak will not grant them access to the web site itself. You must ensure they have access to the WEBTrak site using the Windows Internet Information Services (IIS) Manager, and you should disable anonymous access to the site to ensure they provide their Windows credentials rather than the default IUSER_&lt;SERVERNAME&gt; account.</p>
IP Address	A static IP address. A Report User of this type will authorize anyone accessing the site from a PC with the specified IP address. Note that such users must have access to the web site itself, whether using anonymous access or specific Windows Users and Groups in Windows Internet Information Services (IIS) Manager.

	IP addresses, like computer names, are not secure authorization mechanisms. You should never use IP addresses to authorize users when using DHCP because dynamic IP addresses can change.
Identifier	A unique GUID identifying a single user on a single computer. WEBTrak automatically generates a unique Identifier and saves it to a cookie, which applies to one login on one machine. When a user accesses WEBTrak from another machine they will receive a new Identifier. Refer to the section on User Identifiers for more information.
Computer Name	A computer name. Computer Name type Report Users will authorize anyone accessing WEBTrak from a computer with the specified name, and are the least secure method of authorizing users. Note that users must have access to the site itself, whether using anonymous access or specific Windows Users and Groups in Windows Internet Information Services (IIS) Manager.

After creating the desired Report Users, administrators must grant non-administrator users Browser or Publisher privileges to Report Groups. For more information on Report User roles, refer to the section on Report Groups.

## User Identifiers

This section lists all known User Identifiers, which are unique GUIDs identifying a single user on a single machine. WEBTrak automatically generates a unique Identifier for every login and computer combination and saves it to a cookie on the client machine. These cookies are effectively permanent; they persist a year after the most recent visit to WEBTrak. Identifiers are one type of Report User, which authorize access to WEBTrak.

To streamline the process of creating Report Users using Identifiers, the User Identifiers administration page allows you to quickly create new users based on an Identifier by simply clicking on the Identifier and clicking New User.

## Parameter Types

WEBTrak uses three basic types of parameters, which appear as special controls allowing users to specify Report filters and options.

### Textbox

Textbox parameters allow users to provide validated text or numeric information. In most cases, you can specify Variables for these fields from the Report or Report Type designer.

WEBTrak evaluates variables sequentially from top to bottom, so you can only use non-global variables that appear higher on the designer page. For example, you can use an Event Summary Report's Area variable as part of the Report Title, but not vice versa because WEBTrak evaluates the Area parameter in the General section before evaluating the Report Title parameter in the Report Header section.

Textbox parameter types have the following attributes, which are accessible as Local Variables. You can view these attributes within WEBTrak by hovering over the parameter label in the Report designer.

<b>Textbox Attributes</b>	
Value	The effective value of the parameter, which may come from the user, the Report, or the Report Type.
Caption	The localized label of the parameter.

### **Date Picker**

Date picker parameters provide users with a sophisticated mechanism to select absolute or relative dates and times.

<b>Date Picker Attributes</b>	
Value	The effective value of the parameter, which is a date and time and may come from the user, the Report, or the Report Type.
Caption	The localized label of the parameter.
Date	The date portion of the Value without time.
Time	The time portion of the Value without a date.
Month	The month portion of the Value, which is a number between 1 and 12.
Day	The day portion of the Value, which is a number between 1 and 31 depending on the month.
Year	The four-digit year portion of the Value, such as 2007.
Hour	The hour portion of the Time, which is a number between 0 and 23.
Minute	The minute portion of the Time, which is a number between 0 and 59.
Second	The second portion of the Time, which is a number between 0 and 59.

### **Drop-down**

Drop-down parameters allow users to select predetermined values from a list.

Drop-down parameters have the following attributes, which are accessible as Local Variables. You can view these attributes within WEBTrak by simply hovering over the parameter label in the Report designer.

<b>Drop-Down Attributes</b>	
Value	The effective value of the parameter, which corresponds to the selected drop-down item and may come from the user, the Report, or the Report Type.
Caption	The localized label of the parameter.
Name	The display name of the selected drop-down item, which may come from the user, the Report, or the Report Type.

## **Diagnostics**

The Diagnostics section of WEBTrak provides current status information for clients accessing the site and the server itself. These pages may be useful in configuring and troubleshooting WEBTrak.

## User Information

This diagnostics page identifies the user and machine you are using to access WEBTrak. It provides the following information:

User Information Properties	
User Name	The Windows User login currently being used to access the WEBTrak.
IP Address	The IP address of the PC from which the WEBTrak is being accessed.
Computer Name	The name of the PC from which the WEBTrak is being accessed.
Identifier	The unique User Identifier on the PC from which the WEBTrak is being accessed.
Is Administrator	True if the current session is being accessed by a Report User with administrative rights, False otherwise.

## Server Information

This diagnostics page lists the WEBTrak server's database settings, path, and vendor information. It is only available to Report Users with administrative rights. The following information is available:

Server Information Properties	
Database	The name of the TrakSYS database WEBTrak is using to store its information, such as Report Types, Report Groups, Reports, and Report Users. The default database name is EDB.
Database Server	The name of the server hosting the TrakSYS database.
Vendor	The name of the software vendor who provided you with TrakSYS.
Author	The author of TrakSYS.
Root	The root path of WEBTrak, relative to the server name.

## Report Groups

Report Groups are hierarchical folders containing Reports and other Report Groups. They help organize Reports into a logical structure and specify permissions that control which Report Users can see and access using inheritable Permissions and Variables.

### Properties

This section configures the appearance and other settings for Report Groups.

Report Group Properties	
Parent Report Group	The name of the parent Report Group, or None if this Report Group has no parent.
Key	Unique key identifying this Report Group. This key can be

	used to deep link into the WEBTrak from external sites using special Query String URL parameters.
Auto Expand	When checked, the Report Group will appear expanded in the tree view with its direct children visible. When unchecked the Report Group folder is initially closed and collapsed in the tree view.
Icon	An icon representing this Report Group in the tree view.

## Permissions

Permissions control Report User rights to a specific Report Group. Permissions associate a Report User with a Role for a specific Report Group. WEBTrak has four available Roles: Browser, Publisher, No Access, and Deny.

Report Users with Browser permissions can view the Report Group folder and any of its Reports, but they cannot modify the Report Group or access the Report designer to modify any of its Reports.

Publishers can modify anything within the Report Group, and can even create new Reports based on any of the available Report Types.

Report Users assigned the No Access role cannot see or access the Report Group unless otherwise specified.

Report Users assigned the Deny role cannot see or access the Report Group at all.

Report Users with administrative privileges automatically assume the Publisher Role in every Report Group.

Permissions are inheritable. Child Report Groups can inherit permission settings from their parents by checking the Uses Parent property.

<b>Report Group Permission Properties</b>	
User	The Report User this permission applies to.
Type	The Report User's type, such as User, Group, or Computer Name.
Uses Parent	When True, this permission will inherit its setting from the parent Report Group. When False, it has its own value and overrides any matching permission in a parent Report Group.
Role	Which Role the User has in this Report Group. The Role will be Browser, Publisher, No Access, or Deny.

## Variables

This section allows you to specify values for Report Variables. You can either specify an explicit value for each variable or, if not configuring a top-level Report Group, specify that the variable should inherit its value from the parent Report Group folder.

<b>Report Group Variable Properties</b>	
Key	The corresponding Report Variable key, which uniquely identifies the variable throughout WEBTrak.
Uses Parent	When Yes (checked), the variable will inherit its value from the parent Report Group. When No (unchecked), you must assign a value in the Value property.
Value	The effective value of the Report Variable. When Uses Parent is Yes, this will display the inherited value.

## Reports

Reports are finely-tuned, custom views of base Report Types. Whereas Report Types are typically very generic and have many parameters, Reports usually have much narrower scope and far fewer parameters because they mask the complexity of their underlying Report Types. Reports often target specific Areas, Systems, or KPI Calculations, allowing users to quickly obtain the information they need without repeatedly supplying numerous options and filters.

## Design

The Report Designer allows Report Users with Administrative or Publisher rights to specify parameter values and their visibility to end users of reports. The available parameters depend on the selected Report's base Report Type.

### Date and Time Picker

Date and time parameters in WEBTrak use a sophisticated date and time picker, which allows you to specify an exact date and time or an adjustable date and time relative to when users generate a Report. The simplest use of the picker is to specify an explicit, user-defined date and time.

Alternatively, you can select from one of the predefined, relative dates and times such as "Today" or "Month Start". When using one of the predefined date and time options, you can also specify an offset whose units match those of the selected option. For example, selecting "Today" and specifying an offset of +1 is equivalent to tomorrow. Similarly, selecting "Current Time" and specifying an offset of -3600 is equivalent to an hour ago.

When using a relative date and time option with an otherwise ambiguous time, such as "Today" or "Week Start", you can specify the corresponding time by setting the Default Time field. For example, if you select "Today" and specify "12:00 AM" as the Default Time, the effective date and time will be today at midnight (relative to when a user generates a Report). Alternatively, you might specify "12:00 PM" as the Default Time for "Today", which would make the effective date and time today at noon.











When using the "Week Start" or "Week End" options, you can select what day of the week to use as the start of the week using the First Day of Week dropdown. For example, if you select "Week Start" and select "Monday" as the First Day of Week, the effective date will be Monday of the current week.

The Format field at the bottom of the date and time picker allows you to specify how to format the resulting date and time. This setting uses standard .NET date and time formats; refer to the .NET Class Library Reference for a comprehensive listing for all available formats. The default Format is "G".

### Date and Time Format Reference

Standard .NET Date and Time Format Specifiers	
d	Culture-dependent short date pattern.
D	Culture-dependent long date pattern.
t	Culture-dependent short time pattern.
T	Culture-dependent long time pattern.
f	Culture-dependent full date/time pattern (short time).
F	Culture-dependent full date/time pattern (long time).
g	Culture-dependent general date/time pattern (short time).
G	Culture-dependent general date/time pattern (long time).
R/r	Culture-independent RFC1123 pattern: "ddd, dd MMM yyyy HH:mm:ss G\MT".
s	Culture-independent sortable ISO 8601 date/time pattern: "yyyy-MM-ddTHH:mm:ss".
u	Culture-independent universal sortable date/time pattern using local time: "yyyy-MM-dd HH:mm:ssZ".
U	Culture-dependent universal sortable date/time pattern, converted to universal time.
Y/y	Culture-dependent year month pattern.

### Icons

	This item will be visible to Report Users.
	This item will be invisible to Report Users.
	This Parameter Group will display the expand/collapse (↕/⤴) icons.
	This Parameter Group will not display the expand/collapse (↕/⤴) icons.
	This Parameter Group will appear expanded.
	This Parameter Group will appear collapsed.
	This Parameter Group's header will be visible.
	This Parameter Group's header will be hidden.
	This Parameter or Parameter Group will inherit its value and appearance from the base Report Type.
	This Parameter or Parameter Group will override the base Report Type's default value and appearance.

## Reports

### Common Properties

This section describes the common properties of every Report. Use the Report Designer to view and edit these properties.

General	
Name	Display name of the Report as it will appear in WEBTrak.
Description	Description of the Report.

Auto Open	When checked, the Report will automatically open without prompting users for additional parameters or filters.
Icon	The display icon for the Report.
<b>Advanced</b>	
Notes	Notes for the Report. WEBTrak does not display this field to end users.
Key	Unique alphanumeric key string identifying the Report. If referencing a Report externally (by calling report/view.aspx), this allows you to link to a Report by key rather than its internal ID.
Url	The URL for the report, which is typically a Reporting Services URL.
Target	Setting which determines how the Report will open in the browser. "Same Window" means Reports will open inside WEBTrak. "New Window (Single)" means Reports will open in a new browser window, but will reuse the same new window for each subsequent call to the same Report. "New Window (Multiple)" means each call to the Report will open a new window.
Window	Parameters for the Report browser window, specifying settings for location, size, appearance, and behavior.
Open Caption	Label that appears on the button that launches the Report. This typically uses a variable, such as openCaptionOpen, which is usually "Open".
Report Type	The name of the parent Report Type.
Report Type Key	The unique Key of the parent Report Type.

## Dashboards

Dashboard Reports allow for the custom arrangement of smaller report components called Web Parts.

### Dashboard

This Report is the standard pallet for creating custom dashboard reports. The Design properties of this report allow for different Zone layouts to be specified.

Each dashboard page contains exactly 10 Zones. A Zone is a container that can hold any number of Web Parts. A Zone can have one of two orientations, Horizontal or Vertical. The Zone's orientation is specified by background arrow icons when the Zone is empty. Horizontal Zones will stack Web Parts side by side and Vertical Zones will stack Web Parts up and down.

### Parameters


<b>General</b>	
Mode	View or Edit.  View mode specifies that when the report is clicked in the WEBTrak tree view, the end-user rendering of the report is to be displayed.



	Edit mode specifies that when the report is clicked in the WEBTrak tree view, the editing and arrangement rendering of the report is to be displayed. In this mode, Web Parts may be edited and moved around the different Zones available on the dashboard. This mode is only available to users with the appropriate permissions (Publisher or Administrator).
Type	<p>Shared or User Customizable</p> <p>When set to Shared, the dashboard is displayed the same to all users who can access the page.</p> <p>In User Customizable mode, the Web Parts that are initially populated by the page designer are displayed, but each individual user may add new Web Parts to the page creating their own custom views. Users may not alter or move the original Web Parts and do not have access to the Web Parts that other users have added.</p>
Width	Specifies the width in pixels for the Zone layout pattern.
Show Menu	Determines if the top level WEBTrak menu is displayed on the dashboard.
Show Header	Determines if the standard dashboard page header (including the page title and product logo) is displayed on the dashboard.
Pattern	Specifies the Zone layout pattern. There are several pre-defined patterns and the option Custom for user-defined layouts (see Custom Pattern below).
Custom Pattern	When the Pattern property (above) is set to Custom Pattern, this setting is used to define the desired Zone layout. The Custom Pattern format is a sequence of H (Horizontal) and V (Vertical) characters specifying the Zone layout orientation. The   character is used to specify a new row of Zones. If less than 10 Zones are defined in the format string, the remaining Zones (up to 10) are filled in as one V per row.
<b>Time Refresh Intervals</b>	
Timer N Name	Specifies a descriptive name for each Timer. There are 5 Timers available to each dashboard. Timers are used to trigger individual Web Parts to refresh their contents. Each Web Part can be individually associated with a Timer from this list.
Timer N Refresh Seconds	Specifies the interval (in seconds) that the Timer will trigger Web Part refreshes. If the number of seconds is 0, the Timer is effectively disabled.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.

## Event Reports

These reports display Events recorded by LOGICTrak, organized by Areas, Systems, and other Configuration Entities. Detail views within these reports, accessible via the

 detail icon, display the corresponding Event Definition, Sub-System, operator provided data, and all other information related to an Event.

These reports are an excellent source of information for identifying problematic or under-performing equipment. For OEE and other productivity measurements, refer to the KPI Reports.

### Area Event Summary (Date)

This Report displays Event data in various formats. The primary filters for this report are Area and Start and End dates, which limit the data to events overlapping a date range for a particular Area.

This Report uses the Start and End Date parameters to select Event data based on the EventDate field of the Event data table. LOGICTrak populates this field based on the Event Date property of the Event Definition that triggered the Event.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be _blank, _self (the default), _parent, or _top.
<b>General</b>	
Area	The Area for this Report. This setting narrows the results to a one of the Areas in your configuration.
System View	The System View for this Report. This setting narrows the results to one of the System Views in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Display Type	How to display the Report's results, such as in a Tabular format or one of several charts.

Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
Show Totals	When True, the Report will display grand totals at the bottom.
<b>Schemes</b>	
Category Scheme	The Event Category Scheme to use for labeling Event Category levels in the Report.
Capture Scheme	The System Capture Scheme to use for labeling System Capture Columns.
<b>Grouping</b>	
Group	The base configuration entity to use for hierarchically organizing the data. This may be an Area, System, Shift, Product, or almost anything else.
Group Top N	A limit on the number of Groups to return in the Report. For example, if your Group is System and you set Group Top N to 10, the Report will only include data for the first 10 Systems which meet the filter criteria.
Group Is Expanded	Whether or not the Reports should expand the Groups by default.
Group Totals	Whether or not the Reports should display totals for every Group.
Sub-Group	One of the Configuration Entities to use for sub-grouping in the Report. Sub-Groups group the data an additional level within the parent Groups.
Sub-Group Top N	A limit on the number of Sub-Groups to return in the Report.
Sub-Group Is Expanded	Whether or not the Reports should expand the Sub-Groups by default.
Sub-Group Totals	Whether or not the Reports should display totals for every Sub-Group.
<b>Filters</b>	
Shift Schedule	In conjunction with the Shift History parameter, this filters the Report by the current or a recent Shift.
Shift History	Filters the Report by the current or a recent Shift, according to the selected Shift Schedule filter.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product Set	Filter to limit the Report results to a specific Product Set.
Product	Filter to limit the Report results to a specific Product.
OEE Type	Filter to limit the Report results to a specific OEE Type.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.

Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Minimum Minutes	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Seconds to exclude any Events shorter than the Minimum.
Minimum Seconds	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Minutes to exclude any Events shorter than the Minimum.
Maximum Minutes	The maximum duration Events can last for inclusion in the Report. The Report will combine this setting with Maximum Seconds and exclude any Events exceeding the Maximum.
Maximum Seconds	The maximum duration Events can last for inclusion in the Report. The Report will combine this setting with Maximum Minutes and exclude any Events exceeding the Maximum.
<b>Table Definition</b>	
Column 1	The value constituting the first column in the tabular Report. This is typically one of your Configuration Entities or Event related data.

Column 1 (Format)	String format for displaying Column 1 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 2	The value constituting the second column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 2 (Format)	String format for displaying Column 2 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 3	The value constituting the third column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 3 (Format)	String format for displaying Column 3 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 4	The value constituting the fourth column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 4 (Format)	String format for displaying Column 4 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.
<b>Edit</b>	
Allow Edit	This property specifies whether links that allow direct access to the editor interfaces (Event and KPI Interval) will appear within the Report.
Edit Root Path	Root path to the editor pages. The Reports use this to link to the editor pages.

### System Event Summary (Date)

This Report displays Event data in various formats. The primary filter for this report is System, which limits the data to the selected System.

This Report uses the Start and End Date parameters to select Event data based on the EventDate field of the Event data table. LOGICTrak populates this field based on the Event Date property of the Event Definition that triggered the Event.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection

	string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be _blank, _self (the default), _parent, or _top.
<b>General</b>	
Area	The Area for this Report. This setting narrows the results to a one of the Areas in your configuration.
System	The System for this Report. This setting narrows the results to one of the Systems in your configuration.
Sub System	The Sub System for this Report. This setting narrows the results to one of the Sub-Systems in your configuration.
Event Definition	The Event Definition for this Report. This setting narrows the results to one of the Event Definitions in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Display Type	How to display the Report's results, such as in a Tabular format or one of several charts.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
Show Totals	When True, the Report will display grand totals at the bottom.
<b>Grouping</b>	
Group	The base configuration entity to use for hierarchically organizing the data. This may be an Area, System, Shift, Product, or almost anything else.
Group Top N	A limit on the number of Groups to return in the Report. For example, if your Group is System and you set Group Top N to 10, the Report will only include data for the first 10 Systems which meet the filter criteria.
Group Is Expanded	Whether or not the Reports should expand the Groups by default.
Group Totals	Whether or not the Reports should display totals for

	every Group.
Sub-Group	One of the Configuration Entities to use for sub-grouping in the Report. Sub-Groups group the data an additional level within the parent Groups.
Sub-Group Top N	A limit on the number of Sub-Groups to return in the Report.
Sub-Group Is Expanded	Whether or not the Reports should expand the Sub-Groups by default.
Sub-Group Totals	Whether or not the Reports should display totals for every Sub-Group.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
OEE Type	Filter to limit the Report results to a specific OEE Type.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.

Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Minimum Minutes	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Seconds to exclude any Events shorter than the Minimum.
Minimum Seconds	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Minutes to exclude any Events shorter than the Minimum.
Maximum Minutes	The maximum duration Events can last for inclusion in the Report. The Report will combine this setting with Maximum Seconds and exclude any Events exceeding the Maximum.
Maximum Seconds	The maximum duration Events can last for inclusion in the Report. The Report will combine this setting with Maximum Minutes and exclude any Events exceeding the Maximum.
<b>Table Definition</b>	
Column 1	The value constituting the first column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 1 (Format)	String format for displaying Column 1 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 2	The value constituting the second column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 2 (Format)	String format for displaying Column 2 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 3	The value constituting the third column in the tabular



	Report. This is typically one of your Configuration Entities or Event related data.
Column 3 (Format)	String format for displaying Column 3 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 4	The value constituting the fourth column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 4 (Format)	String format for displaying Column 4 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.
<b>Edit</b>	
Allow Edit	This property specifies whether links that allow direct access to the editor interfaces (Event and KPI Interval) will appear within the Report.
Edit Root Path	Root path to the editor pages. The Reports use this to link to the editor pages.

### Area Event Summary (DateTime)

This Report displays Event data in various formats. The primary filters for this report are Area and Start and End dates, which limit the data to events overlapping a date range for a particular Area.

This Report uses the Start and End Date parameters to select Event data based on the StartDateTime field of the Event data table. LOGICTrak populates this field based on the date and time when the Event starts.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.

Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
Area	The Area for this Report. This setting narrows the results to a one of the Areas in your configuration.
System View	The System View for this Report. This setting narrows the results to one of the System Views in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Display Type	How to display the Report's results, such as in a Tabular format or one of several charts.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
Show Totals	When True, the Report will display grand totals at the bottom.
<b>Schemes</b>	
Category Scheme	The Event Category Scheme to use for labeling Event Category levels in the Report.
Capture Scheme	The System Capture Scheme to use for labeling System Capture Columns.
<b>Grouping</b>	
Group	The base configuration entity to use for hierarchically organizing the data. This may be an Area, System, Shift, Product, or almost anything else.
Group Top N	A limit on the number of Groups to return in the Report. For example, if your Group is System and you set Group Top N to 10, the Report will only include data for the first 10 Systems which meet the filter criteria.
Group Is Expanded	Whether or not the Reports should expand the Groups by default.
Group Totals	Whether or not the Reports should display totals for every Group.
Sub-Group	One of the Configuration Entities to use for sub-grouping in the Report. Sub-Groups group the data an additional level within the parent Groups.
Sub-Group Top N	A limit on the number of Sub-Groups to return in the Report.
Sub-Group Is Expanded	Whether or not the Reports should expand the Sub-Groups by default.
Sub-Group Totals	Whether or not the Reports should display totals for every Sub-Group.

<b>Filters</b>	
Shift Schedule	In conjunction with the Shift History parameter, this filters the Report by the current or a recent Shift.
Shift History	Filters the Report by the current or a recent Shift, according to the selected Shift Schedule filter.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product Set	Filter to limit the Report results to a specific Product Set.
Product	Filter to limit the Report results to a specific Product.
OEE Type	Filter to limit the Report results to a specific OEE Type.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.

Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Minimum Minutes	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Seconds to exclude any Events shorter than the Minimum.
Minimum Seconds	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Minutes to exclude any Events shorter than the Minimum.
Maximum Minutes	The maximum duration Events can last for inclusion in the Report. The Report will combine this setting with Maximum Seconds and exclude any Events exceeding the Maximum.
Maximum Seconds	The maximum duration Events can last for inclusion in the Report. The Report will combine this setting with Maximum Minutes and exclude any Events exceeding the Maximum.
<b>Table Definition</b>	
Column 1	The value constituting the first column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 1 (Format)	String format for displaying Column 1 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 2	The value constituting the second column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 2 (Format)	String format for displaying Column 2 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 3	The value constituting the third column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 3 (Format)	String format for displaying Column 3 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 4	The value constituting the fourth column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 4 (Format)	String format for displaying Column 4 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for

	information on available formats.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.
<b>Edit</b>	
Allow Edit	This property specifies whether links that allow direct access to the editor interfaces (Event and KPI Interval) will appear within the Report.
Edit Root Path	Root path to the editor pages. The Reports use this to link to the editor pages.

### System Event Summary (DateTime)

This Report displays Event data in various formats. The primary filter for this report is System, which limits the data to the selected System.

This Report uses the Start and End Date parameters to select Event data based on the StartDateTime field of the Event data table. LOGICTrak populates this field based on the date and time when the Event starts.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be _blank, _self (the default), _parent, or _top.
<b>General</b>	
Area	The Area for this Report. This setting narrows the results to a one of the Areas in your configuration.
System	The System for this Report. This setting narrows the results to one of the Systems in your configuration.
Sub System	The Sub System for this Report. This setting

	narrows the results to one of the Sub-Systems in your configuration.
Event Definition	The Event Definition for this Report. This setting narrows the results to one of the Event Definitions in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Display Type	How to display the Report's results, such as in a Tabular format or one of several charts.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
Show Totals	When True, the Report will display grand totals at the bottom.
<b>Grouping</b>	
Group	The base configuration entity to use for hierarchically organizing the data. This may be an Area, System, Shift, Product, or almost anything else.
Group Top N	A limit on the number of Groups to return in the Report. For example, if your Group is System and you set Group Top N to 10, the Report will only include data for the first 10 Systems which meet the filter criteria.
Group Is Expanded	Whether or not the Reports should expand the Groups by default.
Group Totals	Whether or not the Reports should display totals for every Group.
Sub-Group	One of the Configuration Entities to use for sub-grouping in the Report. Sub-Groups group the data an additional level within the parent Groups.
Sub-Group Top N	A limit on the number of Sub-Groups to return in the Report.
Sub-Group Is Expanded	Whether or not the Reports should expand the Sub-Groups by default.
Sub-Group Totals	Whether or not the Reports should display totals for every Sub-Group.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift

	Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
OEE Type	Filter to limit the Report results to a specific OEE Type.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data

	to exclude.
Minimum Minutes	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Seconds to exclude any Events shorter than the Minimum.
Minimum Seconds	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Minutes to exclude any Events shorter than the Minimum.
Maximum Minutes	The maximum duration Events can last for inclusion in the Report. The Report will combine this setting with Maximum Seconds and exclude any Events exceeding the Maximum.
Maximum Seconds	The maximum duration Events can last for inclusion in the Report. The Report will combine this setting with Maximum Minutes and exclude any Events exceeding the Maximum.
<b>Table Definition</b>	
Column 1	The value constituting the first column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 1 (Format)	String format for displaying Column 1 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 2	The value constituting the second column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 2 (Format)	String format for displaying Column 2 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 3	The value constituting the third column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 3 (Format)	String format for displaying Column 3 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 4	The value constituting the fourth column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 4 (Format)	String format for displaying Column 4 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.
<b>Edit</b>	



Allow Edit	This property specifies whether links that allow direct access to the editor interfaces (Event and KPI Interval) will appear within the Report.
Edit Root Path	Root path to the editor pages. The Reports use this to link to the editor pages.

### Mean Time Between Failure (Date)

This Report displays Mean Time Between Failure Event data in various formats. The primary filter for this report is System, which limits the data to the selected System.

See the MTBF Type property of Event Definitions is used to determine how Event counts and duration are applied to this report.

Mean Time Between Failure (MTBF) is defined as the total duration of the Report (Start Date to End Date) divided by the total Event count. This makes MTBF the average time between the start of one Event and the start of the next.

Mean Time Running Between Failure (MTrBF) is defined as the total duration of the Report (Start Date to End Date) minus the total amount of Event seconds, divided by the total Event count. This makes MTrBF the average time between the end of one Event and the start of the next.

This Report uses the Start and End Date parameters to select Event data based on the EventDate field of the Event data table. LOGICTrak populates this field based on the Event Date property of the Event Definition that triggered the Event.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be _blank, _self (the default), _parent, or _top.
<b>General</b>	
Area	The Area for this Report. This setting narrows the

	results to a one of the Areas in your configuration.
System View	The System View for this Report. This setting narrows the results to one of the System Views in your configuration.
System	The System for this Report. This setting narrows the results to one of the Systems in your configuration.
Sub System	The Sub System for this Report. This setting narrows the results to one of the Sub-Systems in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Display Type	How to display the Report's results, such as in a Tabular format or one of several charts.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
Show Totals	When True, the Report will display grand totals at the bottom.
<b>Grouping</b>	
Group	The base configuration entity to use for hierarchically organizing the data. This may be an Area, Area Type, System, or System Type.
Group Is Expanded	Whether or not the Reports should expand the Groups by default.
Group Totals	Whether or not the Reports should display totals for every Group.
Sub-Group	One of the Configuration Entities to use for sub-grouping in the Report. Sub-Groups group the data an additional level within the parent Groups.
Sub-Group Is Expanded	Whether or not the Reports should expand the Sub-Groups by default.
Sub-Group Totals	Whether or not the Reports should display totals for every Sub-Group.
<b>Filters</b>	
Minimum Minutes	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Seconds to exclude any Events shorter than the Minimum.
Minimum Seconds	The minimum duration Events must last for inclusion in the Report. The Report will combine this setting with Minimum Minutes to exclude any Events shorter than the Minimum.
Maximum Minutes	The maximum duration Events can last for inclusion in the Report. The Report will combine this setting with Maximum Seconds and exclude any Events exceeding the Maximum.
Maximum Seconds	The maximum duration Events can last for inclusion

	in the Report. The Report will combine this setting with Maximum Minutes and exclude any Events exceeding the Maximum.
<b>Table Definition</b>	
Column 1	The value constituting the first column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 1 (Format)	String format for displaying Column 1 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 2	The value constituting the second column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 2 (Format)	String format for displaying Column 2 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 3	The value constituting the third column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 3 (Format)	String format for displaying Column 3 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
Column 4	The value constituting the fourth column in the tabular Report. This is typically one of your Configuration Entities or Event related data.
Column 4 (Format)	String format for displaying Column 4 values. This setting uses standard .NET string, date, and time formats; refer to the .NET Class Library Reference for information on available formats.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.
<b>Edit</b>	
Allow Edit	This property specifies whether links that allow direct access to the editor interfaces (Event and KPI Interval) will appear within the Report.
Edit Root Path	Root path to the editor pages. The Reports use this to link to the editor pages.

## KPI Reports

These reports calculate and display standard productivity KPIs, including OEE, TEEP, and Takt Time. For OEE, there are additional reports for Availability, Performance, and Quality. The Event Reports are useful in conjunction with these reports to investigate root causes of productivity losses.

## OEE Summary

This Report displays aggregate OEE and related KPI data for the selected date range. It also displays initial production counts (Good, Bad and Total) as well as the counts after adjustment.

This Report uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <u>_blank</u> , <u>_self</u> (the default), <u>_parent</u> , or <u>_top</u> .
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Include Exceptions	Whether or not to include KPI Intervals marked as excluded. Users with the required permissions may exclude corrupt or outlier Intervals from appearing in future reports.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.

Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

## OEE Detail

This Report displays OEE data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Group	The period of time each bar on the chart will represent, such as Intervals, Shifts, or Dates.
Sort	The sort order for the selected grouping. When By Value is selected, the trend points and table are sorted by the aggregated Group value. When By Display Order is selected, the trend points and table are sorted by the Group name or Display Order from the configuration.
Include Exceptions	Whether or not to include KPI Intervals marked as excluded. Users with the required permissions may exclude corrupt or

	outlier Intervals from appearing in future reports.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
X Axis Chart Interval	Indicates the frequency of the X Axis labels that will be rendered on the chart. A value of 0 specifies that the frequency is automatically determined by the report rendering engine. A value of 1 specifies that a label be rendered for every X Axis point whereas a value of 3 would cause the label to be rendered on every third point.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to

	exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

### Availability Detail

This Report Type displays Availability data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.



Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Group	The period of time each bar on the chart will represent, such as Intervals, Shifts, or Dates.
Sort	The sort order for the selected grouping. When By Value is selected, the trend points and table are sorted by the aggregated Group value. When By Display Order is selected, the trend points and table are sorted by the Group name or Display Order from the configuration.
Include Exceptions	Whether or not to include KPI Intervals marked as excluded. Users with the required permissions may exclude corrupt or outlier Intervals from appearing in future reports.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
X Axis Chart Interval	Indicates the frequency of the X Axis labels that will be rendered on the chart. A value of 0 specifies that the frequency is automatically determined by the report rendering engine. A value of 1 specifies that a label be rendered for every X Axis point whereas a value of 3 would cause the label to be rendered on every third point.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.

Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

### Performance Detail

This Report Type displays Performance data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <u>_blank</u> , <u>_self</u> (the default), <u>_parent</u> , or <u>_top</u> .
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Group	The period of time each bar on the chart will represent, such as Intervals, Shifts, or Dates.
Sort	The sort order for the selected grouping. When By Value is selected, the trend points and table are sorted by the aggregated Group value. When By Display Order is selected, the trend points and table are sorted by the Group name or Display Order from the configuration.
Include Exceptions	Whether or not to include KPI Intervals marked as excluded. Users with the required permissions may exclude corrupt or outlier Intervals from appearing in future reports.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
X Axis Chart Interval	Indicates the frequency of the X Axis labels that will be rendered on the chart. A value of 0 specifies that the frequency is automatically determined by the report rendering engine. A value of 1 specifies that a label be rendered for every X Axis point whereas a value of 3 would cause the label to be rendered on every third point.
<b>Filters</b>	

Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to

	include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

### Quality Detail

This Report Type displays Quality data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be _blank, _self (the default), _parent, or _top.
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.

End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Group	The period of time each bar on the chart will represent, such as Intervals, Shifts, or Dates.
Sort	The sort order for the selected grouping. When By Value is selected, the trend points and table are sorted by the aggregated Group value. When By Display Order is selected, the trend points and table are sorted by the Group name or Display Order from the configuration.
Include Exceptions	Whether or not to include KPI Intervals marked as excluded. Users with the required permissions may exclude corrupt or outlier Intervals from appearing in future reports.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
X Axis Chart Interval	Indicates the frequency of the X Axis labels that will be rendered on the chart. A value of 0 specifies that the frequency is automatically determined by the report rendering engine. A value of 1 specifies that a label be rendered for every X Axis point whereas a value of 3 would cause the label to be rendered on every third point.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.

Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

### TEEP Detail

This Report Type displays TEEP data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override

	WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Group	The period of time each bar on the chart will represent, such as Intervals, Shifts, or Dates.
Sort	The sort order for the selected grouping. When By Value is selected, the trend points and table are sorted by the aggregated Group value. When By Display Order is selected, the trend points and table are sorted by the Group name or Display Order from the configuration.
Include Exceptions	Whether or not to include KPI Intervals marked as excluded. Users with the required permissions may exclude corrupt or outlier Intervals from appearing in future reports.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
X Axis Chart Interval	Indicates the frequency of the X Axis labels that will be rendered on the chart. A value of 0 specifies that the frequency is automatically determined by the report rendering engine. A value of 1 specifies that a label be rendered for every X Axis point whereas a value of 3 would cause the label to be rendered on every third point.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.



Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

## Takt Detail

This Report Type displays Takt Time data grouped into the selected period for the selected KPI Calculation. The Report contains two sections.

The first section contains a graph showing data series of the calculated KPI for the selected grouping period, the cumulative KPI value from the beginning of the report period, and the Target and Baseline values configured for the selected KPI Calculation.

The second section contains a tabular representation of the data specified by the Report parameters.

This Report Type uses the Start and End Date parameters to select KPI data based on the IntervalDate field of the OeeInterval data table. LOGICTrak populates this field based on the Interval Date property of the selected KPI Calculation.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Group	The period of time each bar on the chart will represent, such as Intervals, Shifts, or Dates.
Sort	The sort order for the selected grouping. When By Value is selected, the trend points and table are sorted by the aggregated Group value. When By Display Order is selected, the trend points and table are sorted by the Group name or Display Order from the configuration.
Include Exceptions	Whether or not to include KPI Intervals marked as excluded.

	Users with the required permissions may exclude corrupt or outlier Intervals from appearing in future reports.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
X Axis Chart Interval	Indicates the frequency of the X Axis labels that will be rendered on the chart. A value of 0 specifies that the frequency is automatically determined by the report rendering engine. A value of 1 specifies that a label be rendered for every X Axis point whereas a value of 3 would cause the label to be rendered on every third point.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.

Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

### Production Loss

This Report Type displays production data grouped into the selected period for the selected KPI Calculation. The Report contains three sections.

The first section contains a graph showing the total production in relation to the Production and Net Operation targets. The Production target is the number of units the line could produce in the total report time minus System Not Scheduled and Availability Loss Event time. The Net Operation target is the number of units the line could produce in the total report time minus System Not Scheduled Event time.

The second section contains a graph showing the loss cost for each period group. The report breaks the losses down by Availability, Performance, and Quality losses. The Report input parameters specify the cost per unit for each of these loss types.

The third section contains a tabular representation of the data specified by the Report parameters.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate.

	Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be _blank, _self (the default), _parent, or _top.
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Group	The period of time each bar on the chart will represent, such as Intervals, Shifts, or Dates.
Include Exceptions	Whether or not to include KPI Intervals marked as excluded. Users with the required permissions may exclude corrupt or outlier Intervals from appearing in future reports.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
X Axis Chart Interval	The sort order for the selected grouping. When By Value is selected, the trend points and table are sorted by the aggregated Group value. When By Display Order is selected, the trend points and table are sorted by the Group name or Display Order from the configuration.
<b>Cost per Calculation Unit</b>	These settings associate cost with each type of OEE loss. For Availability and Performance Loss, the reports use the theoretical rate to calculate the hypothetical number of units lost to each minute of interrupted production time. For Quality Loss, the reports simply use the total number of Bad units. Using these unit counts, the reports then use these Cost per Calculation Unit settings to calculate the estimated financial impact of Availability, Performance, and Quality Loss. Refer to the section on OEE for more information on these OEE factors.
Availability Loss (Field)	One of the KPI Capture Scheme Columns containing the cost per Availability Loss Calculation Unit. The reports multiply the theoretical rate by the total duration of Availability Loss events to obtain the Availability Loss Calculation Units for the specified date range. If this property is None, the report uses the Constant version

	of the property (below) for the cost calculation.
Availability Loss (Constant)	Fixed cost per Availability Loss Calculation Unit. The reports multiply the theoretical rate by the total duration of Availability Loss events to obtain the Availability Loss Calculation Units for the specified date range.
Performance Loss (Field)	One of the KPI Capture Scheme Columns containing the cost per Performance Loss Calculation Unit. The reports multiply the theoretical rate by the total duration of Performance Loss events to obtain the Performance Loss Calculation Units for the specified date range. If this property is None, the report uses the Constant version of the property (below) for the cost calculation.
Performance Loss (Constant)	Fixed cost per Performance Loss Calculation Unit. The reports multiply the theoretical rate by the total duration of Performance Loss events to obtain the Performance Loss Calculation Units for the specified date range.
Quality Loss (Field)	One of the KPI Capture Scheme Columns containing the cost per Quality Loss Calculation Unit. The reports sum the total number of Bad Calculation Units for the specified date range to use as the Quality Loss Calculation Units. If this property is None, the report uses the Constant version of the property (below) for the cost calculation.
Quality Loss (Constant)	Fixed cost per Quality Loss Calculation Unit. The reports sum the total number of Bad Calculation Units for the specified date range to use as the Quality Loss Calculation Units.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to

	include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

### KPI Comparison Detail

This Report Type allows the side-by-side comparison of OEE data across several KPI Calculations. KPI Calculations can be grouped for selection by using KPI Calculation Views in MODELTrak. The Report contains three sections.

The first section contains a trend graph with one line per KPI Calculation plotted across the grouping option selected. The average OEE for all selected KPI Calculations is displayed at the top of the chart.

The second section is a tabular list of the OEE rollup for each OEE Calculation for the selected KPI Calculation View.

The third section contains a tabular list of the OEE breakdown for each group in the selected grouping. The OEE for each KPI Calculation is broken out within each grouping.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
KPI Calculation View	The KPI Calculation View for this Report. This setting narrows the results to the KPI Calculations specified by a KPI Calculation View in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Group	The period of time each bar on the chart will represent, such as Intervals, Shifts, or Dates.
Include Exceptions	Whether or not to include KPI Intervals marked as excluded. Users with the required permissions may exclude corrupt or outlier Intervals from appearing in future reports.
Show Chart Data Labels	When Yes, labels will appear beside every data point on the chart. When No, these labels will remain hidden.
X Axis Chart Interval	Indicates the frequency of the X Axis labels that will be rendered on the chart. A value of 0 specifies that the frequency is automatically determined by the report rendering engine. A value of 1 specifies that a label be rendered for every X Axis point whereas a value of 3 would



	cause the label to be rendered on every third point.
<b>Schemes</b>	
Capture Scheme	The KPI Capture Scheme to use for labeling KPI Capture Columns.
<b>Filters</b>	
Shift Group	Filter to limit the Report results to a specific Shift Group.
Shift	Filter to limit the Report results to a specific Shift.
Team Group	Filter to limit the Report results to a specific Team Group.
Team	Filter to limit the Report results to a specific Team.
Product Set	Filter to limit the Product selection below to a specific Product Set
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.

Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

## Administration

These reports are primarily troubleshooting tools. They may be particularly helpful during initial deployment when configuration, communication, and security problems are most likely to occur.

## Log Summary

This Report Type displays application log data recorded by individual TrakSYS components. Entries in this report represent informational events, warnings, or errors during execution. All of the applications' log information is available through this report, with the exception of entries occurring while the database is inaccessible.

## Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <u>_blank</u> , <u>_self</u> (the default), <u>_parent</u> , or <u>_top</u> .
<b>General</b>	
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Application	String filter to limit the results to a particular source application. Use asterisks (*) as wildcards.
Host	String filter to limit the results to a particular computer name. Use asterisks (*) as wildcards.
Message	String filter to limit the results to log entries with a particular message. Use asterisks (*) as wildcards.
Category	String filter to limit the results to log entries of a particular

	category. Use asterisks (*) as wildcards.
Type	Numeric filter to limit the results to log entries of a particular type. Valid types are Information (1), Warning (2), and Error (3). Use -1 to include all types.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

### KPI Counter Detail

This Report Type displays a trend for a specified KPI Counter for a specified KPI Calculation over a period of time. There are two graphs as well as a tabular view of the counter data. The first chart is a trend of the actual Counter values. The second is a trend of the Counter changes (delta).

A KPI Counter will only record the detail information available in this report when its "Record Detail" property is enabled in MODELTrak.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be _blank, _self (the default), _parent, or _top.
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
KPI Counter	A KPI Counter belonging to the selected KPI Calculation, which selects the report's KPI Counter. Note that you must explicitly configure the Counter to record detail tracking information by selecting the Record Detail option (in the

	Advanced tab) in MODELTrak.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

### KPI Dump

This Report Type generates an Excel file with all of the KPI Interval records for a selected KPI Calculation over a specified date range.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <u>_blank</u> , <u>_self</u> (the default), <u>_parent</u> , or <u>_top</u> .
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Include Exceptions	When set to No, any KPI Interval records that are marked as an "exception" are filtered out of the results.
<b>Filters</b>	
Shift History	Filter to limit the Report results to a specific Shift History period. The options include Current Shift, Last Shift and 2 Shifts Ago. When using the Shift History filter, care must be taken to set the Start and End Date/Time parameters accordingly as they are still used in the query. For example, to select data for the Last Shift, make sure to set the Start and End Date/Time parameters sufficiently to capture the entire previous Shift.
Shift	Filter to limit the Report results to a specific Shift.

Team	Filter to limit the Report results to a specific Team.
Product	Filter to limit the Report results to a specific Product.
Include Suffix	A suffix to append to each Capture Tag XX include filter.
Exclude Suffix	A suffix to append to each Capture Tag XX exclude filter.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 01 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 02 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 03 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 04 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 05 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 06 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 07 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 08 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 09 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
Capture 10	Filter for this Capture Column, specifying what data to include.
Capture 10 (Exclude)	Filter for this Capture Column, specifying what data to exclude.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

## Data Range

This Report Type is used to list the different ranges of data that exist in the database. Systems and KPI Calculations are listed and the minimum and maximum dates that data has been recorded are displayed for each.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <u>_blank</u> , <u>_self</u> (the default), <u>_parent</u> , or <u>_top</u> .
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

#### Audit Summary

This Report Type lists a summary of the audit records that are captured by the AUDITTrak feature for a specified date range and filters.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this

	report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
Start Date	The start date of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date of the Report's date range. Refer to the section on Date and Time Picker for more information.
Entity ID	An integer filter for Entity ID. The Entity IDs for configuration items can be found by turning on the Show IDs feature in MODELTrak and looking in the title bar of an Entities' Properties dialog.
Entity Name	A wildcard filter for Entity Names such as "L1_FILLER*".
Entity Type Name	A wildcard filter for Entity Type Names such as "Area" or "System".
User 01	A wildcard filter for signee names in the Primary Signature field.
User 02	A wildcard filter for signee names in the Secondary Signature field.
Notes	A wildcard filter for comments made in the Notes field.
Expand in Detail	When set to Yes, the details portion of each row will be expanded automatically when the report is opened.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

## Audit Dump

This Report Type generates an Excel file with all of the audit records for a selected over a specified date range and filters.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this

	report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
Start Date	The start date of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
Entity ID	An integer filter for Entity ID. The Entity IDs for configuration items can be found by turning on the Show IDs feature in MODELTrak and looking in the title bar of an Entities' Properties dialog.
Entity Name	A wildcard filter for Entity Names such as "L1_FILLER*".
Entity Type Name	A wildcard filter for Entity Type Names such as "Area" or "System".
User 01	A wildcard filter for signee names in the Primary Signature field.
User 02	A wildcard filter for signee names in the Secondary Signature field.
Notes	A wildcard filter for comments made in the Notes field.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.

## Journal Summary

This Report Type lists a summary of the Journal records that are entered using the IMPROVETrak Journal entry forms.

### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is



	rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
Start Date	The start date of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date of the Report's date range. Refer to the section on Date and Time Picker for more information
Area	The Area for this Report. This setting narrows the results to a one of the Areas in your configuration.
System	The System for this Report. This setting narrows the results to one of the Systems in your configuration.
Category	Filters the results by the selected Category.
User	A wildcard filtering the results by the specified User string.
Notes	A wildcard filtering the results by the specified Notes.
Expand All Notes	When set to Yes, the Notes portion of each row will be expanded automatically when the report is opened.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.
<b>Edit</b>	
Allow Edit	This property specifies whether links that allow direct access to the editor interfaces (Event and KPI Interval) will appear within the Report.
Edit Root Path	Root path to the editor pages. The Reports use this to link to the editor pages.

## Extensibility Reports

### Link URL

This Report Type facilitates custom entries in the WEBTrak tree view which link to user defined URLs. There are no parameters for this Report Type for end users to select. Reports of this type typically use the Auto Open setting so that the link opens as soon as a user clicks the corresponding node in the WEBTrak tree view.

### Parameters

This report does not have any parameters.

### URL Link Refresh

This Report Type provides auto-refresh support to any URL, which allows Reports to periodically regenerate according to a configurable interval without user intervention.

### Parameters

<b>General</b>	
URL	The URL that will be displayed and automatically refreshed.
Refresh Seconds	The refresh interval for the specified URL expressed in seconds.

## Link Reporting Services

This Report Type facilitates custom entries in the WEBTrak tree view which link to SQL Server Reporting Services Reports.

This Report Type offers several Reporting Services parameters which control how Reporting Services requests and displays these Reports.

The parameters for this Report Type are not typically available to end users. These Reports typically use the Auto Open setting so that the link opens as soon as a user clicks the corresponding node in the WEBTrak tree view.

### Parameters


<b>General</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
View Parameters	When True, Reporting Services will display parameter input controls at the top of the Report, allowing end users to modify every Report parameter.
Page	The target page of the Report. If a Report always has multiple pages, you can set this greater than 1 to open a particular page.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
View Document Map	When True, Reporting Services will display the document map outline for the selected Report.
Document Map ID	The target element ID within the document map.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.

## Editors

The Editor Reports allow access to the data editing interfaces for Event and OEE data. The parameters for these Report Types allow customization of the editor forms.

### Event List

This Report Type displays Events for the selected System and date range. It includes on the parent System event and does not include an expandable section showing its sub-events.


Authorized users can click the  edit link to the left of each row to launch the Event editor dialog, where they can edit all of an Event's recorded attributes or delete the Event entirely.

## Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <u>_blank</u> , <u>_self</u> (the default), <u>_parent</u> , or <u>_top</u> .
<b>General</b>	
Area	The Area for this Report. This setting narrows the results to one of the Areas in your configuration.
System	The System for this Report. This setting narrows the results to one of the Systems in your configuration.
Sub System	The Sub System for this Report. This setting narrows the results to one of the Sub-Systems in your configuration.
Start Date	The start date and time of the Report's date range.
End Date	The end date and time of the Report's date range.
Minimum Seconds	The minimum duration Events must last for inclusion in the Report.
Maximum Seconds	The maximum duration Events can last for inclusion in the Report.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.
<b>Edit</b>	
Edit Root Path	Root path to the editor pages. The Reports use this to link to the editor pages.

## Event List with Sub Events

This Report Type displays Events with expandable sub-events for the selected System and date range.

Authorized users can click the  edit link to the left of each row to launch the Event editor dialog, where they can edit all of an Event's recorded attributes or delete the Event entirely.

## Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be _blank, _self (the default), _parent, or _top.
<b>General</b>	
Area	The Area for this Report. This setting narrows the results to one of the Areas in your configuration.
System	The System for this Report. This setting narrows the results to one of the Systems in your configuration.
Sub System	The Sub System for this Report. This setting narrows the results to one of the Sub-Systems in your configuration.
Start Date	The start date and time of the Report's date range.
End Date	The end date and time of the Report's date range.
Minimum Seconds	The minimum duration Events must last for inclusion in the Report.
Maximum Secends	The maximum duration Events can last for inclusion in the Report.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.
<b>Edit</b>	
Edit Root Path	Root path to the editor pages. The Reports use this to link to the editor pages.

## New Event

This Report Type launches a New Event data entry dialog. The parameters of the Report Type can specify default values for some of the Event attributes. For instance, the Report Type may pre-select the System and Event Definition so that


clicking on the New Event node in the WEBTrak tree view will show an entry form with those attributes already set.


#### Parameters

<b>Advanced</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
Editor URL	URL to the New Event data entry dialog.
Popup	When Yes, the New Event data entry dialog will pop-up in a new browser window. Otherwise it will load in the same window.
<b>General</b>	
Area	The default Area for new Events. This setting pre-selects one of the Areas in your configuration.
System	The default System for new Events. This setting pre-selects one of the Systems in your configuration.
Event Definition	The default Event Definition for new Events. This setting pre-selects one of the Event Definitions in your configuration.
Start Date	The default Start Date and time for new Events.
End Date	The default End Date and time for new Events.
Date	The default Date for new Events.

#### KPI Interval List

This Report Type displays KPI Intervals for the selected KPI Calculation and date range.

Authorized users can click the  edit link to the left of each row can to launch the KPI Interval editor dialog, where then can edit all of a KPI Interval's recorded attributes.

The  adjustment link to the left of each row launches the New KPI Interval Adjustment dialog, allowing users to make counter adjustments to the selected Interval.

#### Parameters

<b>Database</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
<b>Reporting Services</b>	
RDL Name	Name of the Report RDL file in Reporting Services.
View Toolbar	When True, Reporting Services will display its toolbar for navigation, zoom, and downloads at the top of the generated Report.
Output Format	The Reporting Services output format to generate. Select HTML to display the Report as a typical HTML Report in a browser window.
Clear Session	When True, prevents Reporting Services from caching this

	report for subsequent requests.
Page	Specifies the Page that is opened initially when the Report is rendered.
Link Target	The HTML frame link target, which determines where links clicked from within the Report will open. This setting may be <code>_blank</code> , <code>_self</code> (the default), <code>_parent</code> , or <code>_top</code> .
<b>General</b>	
KPI Calculation	The KPI Calculation for this Report. This setting narrows the results to one of the KPI Calculations in your configuration.
Start Date	The start date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range. Refer to the section on Date and Time Picker for more information.
<b>Report Header</b>	
Report Title	The text to display at the top of the Report as the title. This is typically a combination of Variables.
Report Sub-Title	The text to display as the Report's subtitle. This is typically a combination of Variables.
<b>Edit</b>	
Edit Root Path	Root path to the editor pages. The Reports use this to link to the editor pages.

### New KPI Interval Adjustment

This Report Type launches a New KPI Interval Adjustment entry dialog for making counter adjustments to a single Interval. The parameters of the Report Type can specify default values for some of the adjustment attributes. For instance, the Report Type may pre-select the KPI Calculation and Adjusted Counter so that clicking on the New KPI Interval Adjustment node in the WEBTrak tree view will show an entry form with those attributes already set.

#### Parameters

<b>Advanced</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
Editor URL	URL to the New KPI Interval Adjustment entry dialog.
Popup	When Yes, the New KPI Interval Adjustment entry dialog will pop-up in a new browser window. Otherwise it will load in the same window.
<b>General</b>	
KPI Calculation	The default KPI Calculation for new Interval adjustments.
Date	The default Date for new Intervals.
Adjustment Type	The type of adjustments to create. Corrections adjust Good or Bad unit counts without modifying the Total, automatically balancing out the unadjusted units to retain the original Total. Additions adjust Good or Bad unit counts and update the Total units count without modifying the unadjusted units.

	For example, a Correction of -2 Good would increment the Bad by 2 and decrement the Good by 2, retaining the original Total units count. An Addition of -2 Good would decrement both the Good and Total counts by 2, leaving the original Bad units count intact.
Adjusted Input	Which units, Good or Bad, adjustments will affect. Note that if the Adjustment Type is Correction, adjustments will modify both Good and Bad to preserve the original Total units count.
Created By	The user responsible for creating adjustments. By default, this setting is the Global Variable {#User}.
<b>Controls</b>	
Show Date	Whether or not the New KPI Interval Adjustment entry dialog should display the Date property to end users.
Show Adjustment Type	Whether or not the New KPI Interval Adjustment entry dialog should display the Adjustment Type property to end users.
Show Adjusted Input	Whether or not the New KPI Interval Adjustment entry dialog should display the Adjusted Input property to end users.
Show Created By	Whether or not the New KPI Interval Adjustment entry dialog should display the Created By property to end users.
Show Notes	Whether or not the New KPI Interval Adjustment entry dialog should display the Notes property to end users.

### New KPI Range Adjustment

This Report Type launches a New OEE Range Adjustment entry dialog for making counter adjustments to a range of Intervals. The parameters of the Report Type can specify default values for some of the adjustment attributes. For instance, the Report Type can pre-select the KPI Calculation, Adjusted Counter, and Start/End Date Range so that clicking on the New OEE Range Adjustment node in the WEBTrak tree view will show an entry form with those attributes already set.

The entry form generates a table with each of the Intervals that fall within the specified range. After entering a Calculation Units adjustment at the top of the screen, clicking the Recalculate button will evenly distribute the adjustment across the selected Intervals according to the Production Time of each Interval. You can also manually modify each Interval's adjustment if necessary. Refer to the section on KPI Units for more information on Calculation Units.

### Parameters

<b>Advanced</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection

	string.
Editor URL	URL to the New OEE Range Adjustment entry dialog.
Popup	When Yes, the New KPI Interval Adjustment entry dialog will pop-up in a new browser window. Otherwise it will load in the same window.
<b>General</b>	
KPI Calculation	The default KPI Calculation for new Interval adjustments.
Adjustment Type	<p>The type of adjustments to create. Corrections adjust Good or Bad unit counts without modifying the Total, automatically balancing out the unadjusted units to retain the original Total. Additions adjust Good or Bad unit counts and update the Total units count without modifying the unadjusted units.</p> <p>For example, a Correction of -2 Good would increment the Bad by 2 and decrement the Good by 2, retaining the original Total units count. An Addition of -2 Good would decrement both the Good and Total counts by 2, leaving the original Bad units count intact.</p>
Adjusted Input	Which units, Good or Bad, adjustments will affect. Note that if the Adjustment Type is Correction, adjustments will modify both Good and Bad to preserve the original Total units count.
Created By	The user responsible for creating adjustments. By default, this setting is the Global Variable {#User}.
<b>Filters</b>	
Start Date	The start date and time of the Report's date range, which filters the table of adjusted Intervals. Refer to the section on Date and Time Picker for more information.
End Date	The end date and time of the Report's date range, which filters the table of adjusted Intervals. Refer to the section on Date and Time Picker for more information.
Use Dates	Whether or not to filter the Intervals using the Start Date and End Date filters.
Shift Group	The Shift Group for this Report. This setting filters the list of Intervals at the bottom of the New OEE Range Adjustment entry dialog, limiting which Intervals receive adjustments.
Shift	The Shift for this Report. This setting filters the list of Intervals at the bottom of the New OEE Range Adjustment entry dialog, limiting which Intervals receive adjustments.
Team Group	The Team Group for this Report. This setting filters the list of Intervals at the bottom of the New OEE Range Adjustment entry dialog, limiting which Intervals receive adjustments.
Team	The Team for this Report. This setting filters the list of Intervals at the bottom of the New OEE Range Adjustment entry dialog, limiting which Intervals



	receive adjustments.
Product	The Product for this Report. This setting filters the list of Intervals at the bottom of the New OEE Range Adjustment entry dialog, limiting which Intervals receive adjustments.
Capture 01	Filter for this Capture Column, specifying what data to include.
Capture 02	Filter for this Capture Column, specifying what data to include.
Capture 03	Filter for this Capture Column, specifying what data to include.
Capture 04	Filter for this Capture Column, specifying what data to include.
Capture 05	Filter for this Capture Column, specifying what data to include.
Capture 06	Filter for this Capture Column, specifying what data to include.
Capture 07	Filter for this Capture Column, specifying what data to include.
Capture 08	Filter for this Capture Column, specifying what data to include.
Capture 09	Filter for this Capture Column, specifying what data to include.
Capture 10	Filter for this Capture Column, specifying what data to include.
<b>Controls</b>	
Show Adjustment Type	Whether or not the New OEE Range Adjustment entry dialog should display the Adjustment Type property to end users.
Show Adjusted Input	Whether or not the New OEE Range Adjustment entry dialog should display the Adjusted Input property to end users.
Show Created By	Whether or not the New OEE Range Adjustment entry dialog should display the Created By property to end users.
Show Notes	Whether or not the New OEE Range Adjustment entry dialog should display the Notes property to end users.
Show Date	Whether or not the New OEE Range Adjustment entry dialog should display the Date property to end users.
Show Shift	Whether or not the New OEE Range Adjustment entry dialog should display the Shift property to end users.
Show Team	Whether or not the New OEE Range Adjustment entry dialog should display the Team property to end users.
Show Product	Whether or not the New OEE Range Adjustment entry dialog should display the Product property to end users.
Show Capture Values	Whether or not the New OEE Range Adjustment entry dialog should display the Capture Value properties to end users.

### **New Journal Entry**

This Report Type launches an IMPROVETrak Journal data entry dialog.

## Parameters

<b>Advanced</b>	
Connection String	Database connection string for Reports to obtain parameter data. This setting will affect the queries used to populate dropdowns from the database. Use this setting to override WEBTrak's default connection string.
Editor URL	URL to the New Journal entry dialog.
Popup	When Yes, the New Journal entry dialog will pop-up in a new browser window. Otherwise it will load in the same window.
<b>General</b>	
Area	The default Area for new Journal Entry.
System	The default System for new Journal Entry.
Category	The default Journal Category for new Journal Entry.
Date/Time	The default Date for new Journal Entry.

## Variables

Variables in WEBTrak contain shared values. WEBTrak has three types of variables. When variables are used in parameter expressions, they should always be enclosed in curly braces {}. So a variable called #Culture would be used as {#Culture}.

### Global Variables

Global Variables are predefined and available throughout WEBTrak. Their values change depending on the current context, but their meaning is always identical. The table below describes the available global variables.

<b>Global Variables</b>	
#Culture	The root part of the end user's culture setting which indicates their language. For example, "en-US" is the culture for English speaking users in the U.S. The Culture variable returns only the first part of this string "en" (trimming the "-US" part). The Culture is typically part of the path to localized Reports, so this global variable is common in Report URLs. The culture is determined by hidden header information sent to the WEBTrak site by the end user's browser.
#CultureFull	The end user's full culture setting which indicates their language and region. For example, "en-US" is the culture for English speaking users in the U.S. The culture is determined by hidden header information sent to the WEBTrak site by the end user's browser.
#ReportName	The name of the Report. When referenced from a Report Type, this variable corresponds to the name of the Report based on the Report Type.
#ReportGroupName	The name of the Report Group containing the Report.
#ReportDescription	The description for the Report.
#Now	The date and time the user generates the Report.
#User	The name of the Report User generating the Report.
#RootPath	The root path URL of WEBTrak, relative to the server. "/WEBTrak/" is the typical value of this variable.

## Report Variables

Report Variables are available to all Reports, but they receive their values from Report Groups. You define these in the Report Variables section of Administration and then assign them values in the Report Groups.

WEBTrak offers Variable Groups which allow you to group together sets of related Report Variables. Event Report Variable belongs to a Variable Group, and will appear in the same hierarchy within the Report Group variables section. Report Variables inherit their value from an ancestor Report Group if their Uses Parent attribute is "Yes".

## Local Variables

The third type of variables is Local Variables. Report Types define local variables which are common to all derived Reports. You can browse the names of local variables by simply hovering over parameter labels in the Report designer. Refer to the Parameter Types section for more information on common parameter attributes accessible as local variables.

## Report Programming

### Calling Reports by Key

Each Report in that is created in the report Tree View can be assigned a unique Key. The Key is specified in the Advanced section of the Report's Design View. The following URL syntax can be used to link to a report from an external page/site.

`http://<servername>/WEBTrak/report/view.aspx?ReportKey=<key>`

The <servername> is the name of the server where WEBTrak is installed and the <key> is the Key specified for the desired report.

### Special Report Parameters

There are several special report parameters that can be used to control the behavior of the reports when called via URL.

AutoOpen	This parameter is used to launch a report directly into the actual report (bypassing the parameter page). Pass the value of 1 for this parameter to force the report to be opened without allowing parameters to be supplied by the user.
ShowHeaderMenu	The parameter is used to control the visibility of the menu on the parameter page for reports. Pass the value of 0 to hide the menu. This parameter is not used if the AutoOpen parameter is passed as 1 (the parameter page will not be displayed).

For example, the following URL can be used to force the report to automatically open.

`http://<servername>/WEBTrak/report/view.aspx?ReportKey=<key>&AutoOpen=1`

## Localization

The reports are available in several languages. By default, reports are retrieved based on the Regional Settings of the web client PC (automatically passed through the browser and detected). If the Regional Settings of the web client do not match any of the available report languages, the user may see an error message when the report is displayed. In this case, a report configuration change is necessary to retrieve the default English reports.

Reports are stored in SQL Server Reporting Services in a folder structure by language. English reports are in a folder called "en", French reports in a folder called "fr" etc... The URLs for each Report Type utilize a Report Variable called {culture} which in turn references the global {#Culture} variable to dynamically determine the web clients Regional Settings and therefore the Reporting Services folder to retrieve the correct report. To change the language folder for all of the reports to English reports, the {culture} Report Variable in the root Report Group should be changed to "en".

## Dashboard Reports

Dashboards are a special type of reporting interface that allows data views to be constructed using smaller, specialized parts called Web Parts. Web Parts can be charts, small pieces of data, tables or whatever each Web Part exposes. WEBTrak allows Dashboard reports to be created using the Dashboard Report Type. Dashboard reports are different than the other standard reports in TrakSYS as they do not use the SQL Server Reporting Services engine to render reports. The following sections detail the special features and use of Dashboard Reporting.

### View vs. Edit Mode


Dashboard reports can be displayed in one of two modes, View and Edit. View mode is when the report is displayed as intended for end users. Web Parts retrieve their data from the configured sources and render their contents. Edit mode is used for editing the Dashboard layout, adding, removing, moving and configuring Web Parts and their properties. Edit mode can only be entered on a Dashboard when the WEBTrak user has Publisher or Administrator privileges on the folder that contains the report. In addition, Dashboard reports can only be edited (Edit Mode), when WEBTrak users can be identified. In other words, Dashboard reports cannot be edited when the WEBTrak site is configured for anonymous access and Windows authentication is used.




### Zones

Each Dashboard report is a pallet where individual Web Parts can be arranged to display the desired data. A Dashboard is made up of 10 Zones. Each Zone can be configured with a Vertical or Horizontal orientation and can hold any number of Web Parts. Vertical Zones stack Web Parts one on top of each other whereas Horizontal Zones stack Web Parts side by side. The Zone layout for each Dashboard Report can be configured in the Design view for the report.

## Web Part Basics

### Web Part Buttons

	Copy	Copies the Web Part into the clipboard for pasting.
-------------------------------------------------------------------------------------	------	-----------------------------------------------------

	Properties	Opens the Properties page for the selected Web Part.
	Collapse/Expand	Hides or shows the entire Web Part contents below its header.
	Delete	Permanently removes the selected Web Part from the page.

## Adding Web Parts

When the Dashboard is in Edit mode, Web Parts can be added to any Zone by clicking the Add button at the top of the Zone border. This displays a Web Part catalog dialog. Selecting a Web Part from the catalog adds it to the Dashboard Zone.

## Moving Web Parts

In Dashboard Edit mode, Web Parts can be moved within Zones or from one Zone to another by clicking the Part's header and dragging it to the desired location.

## Copying and Pasting Web Parts

In Dashboard Edit mode, Web Parts can be copied into the clipboard by clicking its Copy button in its header. Once a Web Part is copied, it can be pasted into a Zone on the same page (or on another) by clicking the Paste button in the top right corner of the desired Zone.

## Common Web Part Properties (Appearance)

Most Web Parts share a common set of properties that control the basic appearance of the part. These properties are described below.

Common Properties	
Title	Specifies the title that will be displayed in the Web Part's header band.
Border Style	Specifies the style of the outer border and header for the Web Part.
Width x Height	Specifies the outer physical dimensions of the Web Part as it is displayed on the page. Of note is that Web Parts placed in Vertically orientated Zones will always have their Width expanded to 100% of the Zone regardless of the value specified here.
Common Chart Properties	
Chart Title	Specifies the title that will show within the chart area of the Web Part.
Chart Border	Specifies the style of the outer chart border.
Chart Legend Visible	Specifies whether or not the chart legend is displayed in the Web Part.

## Common Web Part Properties (Source)

Most Web Parts share a common set of properties that specify its data source. These properties are described below.

Source Properties	
Mode	Specifies the data source. Options are [Default], SQL or Excel. When the Mode is set to [Default],

	the Web Part retrieves the data based on the properties set in its Data section.
Data Table Source	When the Mode (above) is set to [Default], the Web Parts default query can be overridden by selecting another valid data source on the dashboard page (SQL or Excel Data Web Parts).
Connection String	When the Mode (above) is set to SQL, this property specifies the OLEDB connection string for the query specified by the SQL property (below).
SQL Query	When the Mode (above) is set to SQL, this property specifies the query that will be used to retrieve the data table for the Web Part. The columns in the data table retrieved must conform to the Web Part's expected data interface.
Excel File	When the Mode (above) is set to Excel, this property specifies the workbook that will be accessed to retrieve the Web Part's data table.
Excel Sheet	When the Mode (above) is set to Excel, this property specifies the worksheet (within the selected workbook) that will be accessed to retrieve the Web Part's data table. The columns in the worksheet must conform to the Web Part's expected data interface.
Refresh Timer	Specifies the Refresh Timer (configured in the Dashboard Report Type design mode) that is used to refresh the data source and Web Part. If the selected Refresh Timer is configured to an interval other than 0, the Web Part will refresh its contents at this interval.

### **Date/Time and Filter Source Web Part Properties (Data)**

The Date/Time and Filter Source Web Part properties are used to reference other Web Parts on the same dashboard page. This allows many Web Parts on the same page to use the same filter sources when retrieving data. The referenced filter parts are only used when the Web Part is using its own internal data table query (not when set to SQL or Excel).

### **Custom Filter Web Part Property (Data)**

The Custom Filter property (available on selected Event and KPI Web Parts) allows additional filters to be applied to the standard built in web part queries. For example, specifying **JobID=3** in the Custom Filter field for an Event Bar Chart Web Part would apply an additional filter to the built in query and return only events that were recorded for a Job ID 3. Multiple filter values can be specified by listing each filter=value pair on its own line. The available fields for each Web Part can be found in separate online documentation.

### **Property Mapping Web Part Property (Advanced)**

The Property Mapping property allows for any of a Web Parts design time parameters, to be overridden with run time values. The tooltip section in the content editor lists all of the available Web Part properties that can be assigned using Property Mapping. For example, specifying **SystemID={qs|SystemIID}** would

assign the query string variable called SystemID, to the Web Part's SystemID property. Values assigned in the Property Mapping always take precedence over the design time configured Web Part parameters.

## Contentable Web Part Properties

Certain Web Part properties allow the use of special expression syntax to access a variety of data source values. The properties are generically referred to as "contentable properties". Content can contain a combination of text and any number of Content Expressions. The Expression syntax in Content Editors is as follows

```
{<type><escape>|<identifier>:<format>|<conditions>|<else>}
```

### <type> (Required)

Possible values are:

#### *constant*

Used to for testing. The specified value following the type is returned as is to the calling context.

#### *data*

Used to retrieve data from a table data source. The specific row number in the table data source is specified by adding [] with the zero-based row number (data[2] = the 3rd row).

#### *row*

Used to retrieve data from the current row of a table data source. The row type is only valid in certain contexts.

#### *param*

Used to retrieve data from a parameter of the current web part.

#### *tag*

Used to retrieve the current value from a TrakSYS Tag. The value is retrieved from the database.

#### *qs*

Used to retrieve values from the web page Query String collection.

#### *form*

Used to retrieve values from the web page Forms collection.

#### *datetime*

Used to retrieve current and calculated date/time values.

#### *entity*

Used to retrieve properties of TrakSYS configuration entities.

#### *context*

Used to retrieve values from pre-defined functions.

#### *sequence*

Used to retrieve the next possible unique integer ID for TrakSYS database entities. The sequence type can only be used in the post-back HTML SQL content.

#### **<escape> (Optional)**

Determines the characters that are escaped in the return value. Possible values are .js (JavaScript), .url (URL), .html (HTML) or .sql (SQL).

#### **<identifier> (Required)**

When the *<type>* is *data* or *row*, this is the name of the field from the relevant data source. When the *<type>* is *param*, this is the name of the relevant property (typically Web Part property). When the *<type>* is *tag*, this is the name of a Tag from the configuration. When the *<type>* is *qs* or *form*, this is the name of the web page query string or web page forms collection parameter.

When the *<type>* is *data* or *row* and the data table contains fields named StartDateTime and EndDateTime, the identifier suffix *.TimeSpanDifference* can be used to return a formatted string containing the hours, minutes and seconds of the difference (hh:mm:ss). If the start and end fields have prefixes such as EventStartDateTime and EventEndDateTime, the prefix can be included in the identifier as follows Event.TimeSpanDifference.

When the *<type>* is *data* or *row* and the data table contain a field that returns a duration in terms of seconds, the identifier suffix *.TimeSpanSeconds* can be used to return a formatted string containing the hours, minutes and seconds of the duration (hh:mm:ss). If the source field is in terms of Minutes or Hours, the suffix *.TimeSpanMinutes* or *.TimeSpanHours* can also be used. In any of these cases if a format string is applied, the return value is formatted based on the specified format rather than the default hh:mm:ss. In addition a source field can be converted from one unit of time measurement to another by adding an additional suffix of *.Seconds*, *.Minutes* or *.Hours*. For example, a data source field called EventSeconds can be converted to hours and displayed with one decimal point of precision with the following Expression {row|EventSeconds.TimeSpanSeconds.Hours:N1}.

#### **<format> (Optional)**

This is a .NET format string that is applied to the value of the content after it is calculated.

#### **<conditions> (Optional)**

The conditions are evaluated in the order listed. The first condition that matches the *<identifier>*'s value will return its result as the value of the expression. If there are no conditions specified, then identifier's value is returned as is. Conditions are separated with a semi-colon (;). There are four types of conditions, value:result;, [from\_to]:result;, [> < >= <= <>value]:result; and [null]:result;.

#### **<else> (Optional)**

This is the constant value returned if none of the conditions (above) evaluate to True.

### **Expression Examples**

#### ***constant***



*{constant|red}*

Returns the string "red".

### **data**

*{data[0]|SystemName}*

Returns the value of the column called SystemName from the first row in the source data table.

### **row**

*{row|SystemName}*

Returns the value of the column called SystemName from the current row in the source data table.

*{row|Event.TimeSpanDifference}*

Returns the formatted value (hh:mm:ss) of the difference between the columns EventStartDateTime and EventEndDateTime from the current row in the source data table.

*{row|EventDurationSeconds.TimeSpanSeconds}*

Returns the formatted value (hh:mm:ss) of the number of seconds from the column called EventDurationSeconds from the current row in the source data table.

*{row|EventDurationSeconds.TimeSpanSeconds.Hours:N1}*

Returns the number of hours from the column called EventDurationSeconds (formatted to 1 decimal place) from the current row in the source data table.

### **param**

*{param|AreaID}*

Returns the value of the web part parameter called AreaID from parent web part.

### **tag**

*{tag|BL1.FILLER.TEMP:N2}*

Returns the value of the Tag called BL1.FILLER.TEMP formatted to 2 decimal places.

*{tag|BL1.FILLER.TEMP|[25\_50]:orange;[51\_100]:red;|yellow}*

Returns the string value orange if the BL1.FILLER.TEMP Tag value is between 0 and 50, red if the value is between 51 and 100, and yellow if it is any other value.

*{tag|BL1.FILLER.TEMP|[>50]:high;|normal}*

Returns the string value high if the BL1.FILLER.TEMP Tag value is greater than 50 and normal if it is 50 or less.

### **qs**

*{qs|StartDateTime}*

Returns the value of the html query string parameter called StartDateime from the dashboard page URL.

*{qs.sql|Notes}*

Returns the value of the html query string parameter called Notes from the dashboard page URL. The return value will be escaped based on SQL Server syntax. For example if the Notes value contains a single quote such as It was Joe's Fault, the return value would be It was Joe's Fault.

### **form**

*{form|StartDateTime}*

Returns the value of the html forms collection parameter called StartDateTime from the dashboard page postback.

*{form|AreaID|[/null]:-1; }*

Returns the value of the html forms collection parameter called AreaID from the dashboard page postback. If the parameter does not exist in the collection (returns a null value), then the value of -1 will be returned instead.

### **datetime**

*{datetime|Time}* : Returns the current time only.

*{datetime|Date}* : Returns the current date (at 12:00:00).

*{datetime|DateStart}* : Returns the current date (at 12:00:00).

*{datetime|DateEnd}* : Returns the current date (at 23:59:59).

*{datetime|MinuteStart}* : Returns the current date/time at the start of the current minute.

*{datetime|MinuteEnd}* : Returns the current date/time at the end of the current minute.

*{datetime|HourStart}* : Returns the current date/time at the start of the current hour.

*{datetime|HourEnd}* : Returns the current date/time at the end of the current hour.

*{datetime|WeekStart}* : Returns the current date/time at the start of the current week.

*{datetime|WeekEnd}* : Returns the current date/time at the end of the current week.

*{datetime|MonthStart}* : Returns the current date/time at the start of the current month.

*{datetime|MonthEnd}* : Returns the current date/time at the end of the current month.

*{datetime|YearStart}* : Returns the current date/time at the start of the current year.

*{datetime|YearEnd}* : Returns the current date/time at the end of the current year.

*{datetime|Date.AddHours(2).AddMinutes(-15).AddSeconds(3)}*

Returns the current date and adds 2 hours, subtracts 15 minutes and adds 3 seconds.

### **entity**

*{entity[3]|System.Name}*

Returns the Name property from the System with ID 3.

*{entity[form.SystemID]|System.Name}*

Returns the Name property from the System with the ID specified by the web page Form collection variable called SystemID.

*{entity[qs.SystemID]|System.Name}*

Returns the Name property from the System with the ID specified by the web page Query String collection variable called SystemID.

*{entity[qs.SystemID]|System.CurrentJobID}*

Returns the Job ID property for the System with the ID specified by the web page Query String collection variable called SystemID.

*{entity[qs.SystemID]|System.CurrentJobName}*

Returns the Job Name property for the System with the ID specified by the web page Query String collection variable called SystemID.

### **context**

*{context|User}*

Returns the user name for the currently logged in WEBTrak user.

*{context|ReportID}*

Returns the ID for the current Report selected in the WEBTrak tree view.

*{context|ReportName}*

Returns the Name for the current Report selected in the WEBTrak tree view.

*{context|ReportKey}*

Returns the Key for the current Report selected in the WEBTrak tree view.

*{context|Culture}*

Returns the shortened culture string passed to the server by the client browser. If the full culture string is "en-US", then this expression would return "en".

*{context|CultureFull}*

Returns the full culture string passed to the server by the client browser (for example "en-US").

*{context|RootPath}*

Returns root path to the current WEBTrak page (for example "http://servername/webtrak/").

### **sequence**

*{sequence|EventID}*

Returns the next available ID for the tEvent table.

## **Web Parts**

This section will describe the standard Web Parts available in TrakSYS.

### **Event and KPI Charts**

The Event and KPI charts focus on displaying data related to captured Event Definitions and built in KPIs like OEE and its associated components.

## **Event Charts**

### **Event Pie Chart**

The Event Pie Chart displays the events from one or more Systems as a percentage of the whole. Events can be grouped by various options and the Pie Chart can be configured to calculate slices based on event time or event count.

### **Event Funnel Chart**

The Event Funnel Chart displays the events from one or more Systems as a percentage of the whole. Events can be grouped by various options and the Funnel Chart can be configured to calculate slices based on event time or event count.

### **Event Pyramid Chart**

The Event Pyramid Chart displays the events from one or more Systems as a percentage of the whole. Events can be grouped by various options and the Pyramid Chart can be configured to calculate slices based on event time or event count.

### **Event Doughnut Chart**

The Event Doughnut Chart displays the events from one or more Systems as a percentage of the whole. Events can be grouped by various options and the Doughnut Chart can be configured to calculate slices based on event time or event count.

### **Event Bar Chart/Column Chart/Pareto Chart**

The Event Bar Chart displays the events from one or more Systems as a series of bars or columns. Events can be grouped by various options and the Bar Chart can be configured to calculate bar height based on event time or event count.

There are three modes for the Event Bar Chart: Bar (horizontal bars), Column (vertical bars) or Pareto. The Pareto option sorts the vertical bars by length and adds a cumulative percentage line to the chart.

### **Event Histogram Chart**

The Event Histogram Chart displays the distribution of event occurrences for different ranges of event durations.

### **Event State Chart**

The Event State Chart displays the events from one or more Systems chronologically in horizontal bars and slices. Each slice in a bar represents a single event from the TrakSYS database.

## **KPI Charts**

### **KPI Trend Chart (Single)**

The KPI Trend (Single) chart displays a calculated KPI over time for a selected KPI Calculation. The selected KPI is plotted as a bar for each grouping interval. The KPI's running average over the reporting period is plotted as a line. In addition, the upper and lower KPI limits/targets can optionally be displayed.

### **KPI Trend Chart (Side by Side)**

The KPI Trend (Side by Side) chart displays a calculated KPI over time for a selected group of KPI Calculations (based on selecting a configured KPI Calculation View).

The selected KPI is plotted as a bar for each KPI Calculation in each grouping interval. The straight KPI average of each grouping interval is plotted as a line.

#### **KPI Trend Chart (Stacked Bar)**

The KPI Trend (Stacked Bar) chart displays a calculated KPI over time for a selected group of KPI Calculations (based on selecting a configured KPI Calculation View). The selected KPI is plotted as a stacked bar for each KPI Calculation in each grouping interval. The straight KPI average of each grouping interval is plotted as a line.

#### **KPI Line Chart**

The KPI Line chart displays several KPI line trends for a selected KPI Calculation over a specified period of time. The KPI set that is plotted can be selected a chart parameter.

#### **KPI Area Chart**

The KPI Area chart displays several KPI line trends for a selected KPI Calculation over a specified period of time. The KPI set that is plotted can be selected a chart parameter. The area under each line appears colored based on the KPI legend color.

#### **KPI Stacked Line Chart**

The KPI Stacked Line chart displays several KPI line trends for a selected KPI Calculation over a specified period of time. The KPI set that is plotted can be selected a chart parameter. Unlike the KPI Line chart, the Stacked variety plots the value of certain non-percentage KPI sets as differentials rather than constant values.

#### **KPI Stacked Area Chart**

The KPI Area chart displays several KPI line trends for a selected KPI Calculation over a specified period of time. The KPI set that is plotted can be selected a chart parameter. The area under each line appears colored based on the KPI legend color. Unlike the KPI Area chart, the Stacked variety plots the value of certain non-percentage KPI sets as differentials rather than constant values.

#### **KPI Radar Chart**

The KPI Radar chart displays several aggregate KPI values for a selected KPI Calculation over a specified period of time. The KPI set that is plotted can be selected a chart parameter. Each KPI value is plotted on its own axis radiating from a common chart center.

#### **KPI Scatter Chart**

The KPI Scatter chart plots a selected KPI vs. the occurrences (count or duration) of a selected Event Definition over a specified date range. An optional linear regression best fit line can be auto calculated and plotted to establish possible correlation.

#### **KPI Gauges**

##### **KPI Radial Gauge**

The KPI Gauge displays a single KPI value from a selected KPI Calculation as a graphical speedometer type gauge. The scale and colored range markers can be configured to produce a variety of different gauge types.

##### **KPI Linear Gauge (Vertical or Horizontal)**

The KPI Gauge displays a single KPI value from a selected KPI Calculation in a horizontal or vertical bar type gauge. The scale and colored range markers can be configured to produce a variety of different gauge presentations.

#### **KPI Production Progress Gauge**

The Production Progress gauge displays the current, best case and projected production counts or times for a selected System and the current Job or Shift.

### **Other Charts and Data**

#### **Historian Parts**

##### **Tag Comparison**

The Tag Comparison chart displays and trends up to 6 historically captured Tag values. There are two separate Y axis scales that allow the first 2 Tags to be scaled to their actual values. The remaining Tags are scaled to the left Y axis. Optionally the upper and lower limits for Tag 1 (as configured in MODELTrak) can be displayed on the chart.

##### **KPI vs. Tag Comparison**

The KPI vs. Tag Comparison chart displays a calculated KPI as a step trend over time for a selected KPI Calculation. The right Y axis plots a selected Tag's historical values over the same period.

#### **SPC Parts**

##### **XBar (R and S) Control Chart**

The XBar (R and S) Control Chart displays tabular and graphical SPC data for a selected Sample Definition over the desired report period. The chart has two mode, XBar-R (Range) and XBar-S (Sigma) which determine the contents of the lower chart trend.

#### **Event Data Parts**

##### **Event Group (Tabular)**

The Event Group Part displays the aggregation of events from one or more Systems in a textual tabular format. Events can be grouped by various options and the rows in the table can be configured to popup or "drill-down" to other report or dashboard pages.

##### **Event List (Tabular)**

The Event List Part displays a list of individual events from one or more Systems in a textual tabular format. The columns displayed in the table can be configured to show various values from the event data and the rows in the table can be configured to popup or "drill-down" to other report or dashboard pages.

##### **Event Detail (Tabular)**

The Event Detail Part displays all of the attributes of a single recorded event.

#### **Content**

##### **HTML Content**

The HTML Content or Value Stream Map allows for ad-hoc HTML content to be created and displayed within dashboard reports. Data can be inserted into the HTML from the parts configured data source (see Data Integration in the HTML Content Web Part below). The HTML Content contains a web based HTML editing component allowing WYSIWYG content creation as well as access to the raw markup for more advanced tasks.

### **Web Frame**

The Web Frame allows for a frame to be inserted onto the dashboard for exposing external (to TrakSYS) web content. The Web Frame will render the web page specified by the URL property of the Web Part.

## **Date and Filter Picker Parts**

### **Date/Time Picker (Calculated)**

The Date/Time Picker (Calculated) allows for user selection of relative date ranges. Instead of selecting independent start and end date/times, the user selects from a dropdown which includes options like "Current Date" or "Last Shift". This Web Part exposes the selected date range to other Web Parts on the page for use in filtering their own data queries.

### **Date/Time Picker (Fixed)**

The Date/Time Picker (Fixed) allows for user selection of fixed date ranges. Users select distinct start and end date/time values. This Web Part exposes the selected date range to other Web Parts on the page for use in filtering their own data queries.

### **Date/Time Picker (Band)**

The Date/Time Picker (Band) allows for user selection of dates using a popup calendar as well as manual entry. It has a thin horizontal footprint allowing it to be efficiently placed at the top of dashboard pages.

### **Date/Time Picker (Query)**

The Date/Time Picker (Query) allows for the specification of a date and time range that can be used by other web parts. The date and time range from this part is specified by a custom SQL query. The query specified must return a specific set of fields that are listed in the content editor tip for the SQL property of this web part.

### **Filter Picker**

The Filter Picker allows for the user selection of several different common report filter criteria. Options include Product, Shift, Team, Job Name, Batch Name, and the various Capture fields. This Web Part exposes the selected filter criteria to other Web Parts on the page for use in filtering their own data queries.

## **Entity Picker Parts**

### **Event Category Picker**

The Event Category Picker displays the Event Categories that are configured for a specified Event Definition. Clicking on an Event Category in the part will display the next level of Categories (if there is a next level configured). The presentation for user selection can be customized by altering the HTML for the part.

### **Product Picker**

The Product Picker displays the Products (and Product Groups) that are configured for a specified Product Set. Clicking on a Product Group in the part will display the next level of Products below that group. The presentation for user selection can be customized by altering the HTML for the part.

## **EVENTTrak Parts**

The EVENTTrak parts require the node that they are displayed on to be configured as a valid enabled EVENTTrak node.

### **Content**

#### **HTML Content (with Post Back)**

The HTML Content (with Post Back) allows for ad-hoc HTML content to be created and displayed within dashboard reports. Data can be inserted into the HTML from the parts configured data source (see Data Integration in the HTML Content Web Part below). The HTML Content contains a web based HTML editing component allowing WYSIWYG content creation as well as access to the raw markup for more advanced tasks.

In addition to the flexible layout and display options, the Post Back enabled HTML part allows for custom SQL to be added to the post back action when the page containing the part is submitted to the server. Values in HTML form elements are available to the SQL statement enabling the creation of complex user interactions which require updates and inserts to the database.

### **Event Parts**

#### **Event List (EVENTTrak)**

The EVENTTrak Event List part displays the active Events for a specified EVENTTrak node. Clicking on one of the events in the list opens a dashboard report specified by the Link Report Key parameter. The presentation for the Event List can be customized by altering the HTML for the part.

#### **Event Data (EVENTTrak)**

The EVENTTrak Event Data part is a specialized version of the HTML Post Back web part. It can be used to allow users to edit certain properties of a selected Event from the EVENTTrak Event List part. On form post back, the web part attempts to locate certain form field values and uses them to construct a built in UPDATE to the database. This part also handles the special save and acknowledge actions for EVENTTrak Events. The presentation for the Event List can be customized by altering the HTML for the part.

### **User Parts**

#### **User Login**

The User Login part is a specialized version of the HTML Post Back web part. It can be used to allow users to login and out at a specified production asset (System). On form post back, the web part attempts to locate certain form field values and uses them to construct built in INSERT and UPDATES to the database. The presentation for the User Login part can be customized by altering the HTML for the part.



## **Miscellaneous**

### **Navigation Parts**

#### **Tabs**

The Tabs part renders a tabbed navigation strip automatically based on the Reports and Report Groups that surround the current page. The two modes (set by the Type web part parameter) for the Tabs part allow for the Reports or Groups and Reports to be included on the Tag. When Reports is selected, only the other direct Report sibling nodes in the tree are rendered as large tabs at the top edge of the part. When Groups and Reports is selected, the direct sibling Report Groups are rendered as the top tabs. The direct Report siblings for the each of the Report Groups are rendered as sub tabs across the bottom of the part.

### **Data Source Parts**

#### **SQL Data Source**

The SQL Data Source allows for the specification of a user defined connection string and SQL query. The results of the query can be exposed to other Web Parts on the page for use as their own custom data source.

#### **Excel Data Source**

The Excel Data Source allows for the specification of an Excel file and Sheet name. The sheet specified must contain a tabular data set. The data set is transformed to a data table and can be exposed to other Web Parts on the page for use as their own custom data source.

### **Tag Parts**

#### **Tag Changer (General)**

The Tag Changer (General) allows for end user manipulation of TrakSYS Virtual Tag values. This part may be used to set values to any Virtual Tags of data type. The specified Tag's value is displayed in a textbox. Changing the value and clicking the update button saves the new value to the TrakSYS database.

#### **Tag Changer (Discrete)**

The Tag Changer (Discrete) allows for end user manipulation of discrete TrakSYS Virtual Tag values. This part may be used to read and set values to any discrete data typed Virtual Tag. The Tags value as displayed by the Web Part can be automatically refreshed periodically using the Refresh Timer setting. Changing the Tag value saves the new value to the TrakSYS database.

## **Data Integration in the HTML Content Web Part**

In addition to creating text and graphical layouts using the HTML Content Web Part, data from a variety of sources can be integrated into the output using the Content Expression syntax (described above for Web Part contentable parameters).

### **Contentable Tags**

The <div>, <td>, <a> and <img> tags contain a special TrakSYS tab in their property pages in the HTML editor. In the TrakSYS tabs, certain HTML attributes and styles can be set dynamically using the Content Expression language.

## Value Replacement

The <div> and <td> tag's contain 10 special Content Expression replacement areas from the TrakSYS tabs in the property pages; Value 1 through Value 10. The results for each Content Expression in the Value properties is placed into the corresponding replacement token in the inner HTML for the tag. For instance, specifying {tag|FILLER.TEMP} for the Value 1 property in a <div> will cause the current value for the FILLER.TEMP Tag into the HTML within the <div> where the {1} token is found.

## ALERTTrak

ALERTTrak is the messaging and notification services for TrakSYS. Notifications can be configured in MODELTrak based on event occurrences, value changes and KPI limits. Messages can be sent to subscribers via email or SMS (text message) formats to PCs, PDAs, pagers and mobile phones.

## ALERTTrak Setup

ALERTTrak is built on Microsoft SQL Server Notification Services and requires SQL 2005 in order to be used. There are a few additional steps required for ALERTTrak functionality to be activated and utilized in TrakSYS.

## SQL Server 2005 Notification Services

The TrakSYS setup for SQL Server 2005 installs the Notification Services option. If another SQL Server setup DVD is used for installation, the Notification Services option must be checked and installed for ALERTTrak to be functional.

## Create Notification Instance

Once TrakSYS is installed and a TrakSYS database has been created, a Notification Instance must be created on the SQL Server. The Database Manager menu option **ALERTTrak | Create Notification Instance** can be used to deploy the necessary backend for ALERTTrak.

## ALERTTrak Configuration

ALERTTrak Subscribers, Devices and Subscriptions can be created and managed via the ALERTTrak section in MODELTrak. If a valid SQL Server with Notification Services installed is not present, the ALERTTrak section will not be displayed in MODELTrak.

## ALERTTrak Execution

The notification message execution is managed by LOGICTrak and SQL Server Notification Services. Message events are created by LOGICTrak based on the configuration and sent to Notification Services where the Subscription rules are evaluated and processed.

## SCRIBETrak

SCRIBETrak is the manual data entry interface for TrakSYS. It is a configurable set of forms enabling the collection of data from assets where automated acquisition is not possible or cost prohibitive. SCRIBETrak allows the productivity related data (e.g. events, counts, poor quality reasons, observation, etc...) to be entered at the

end of the day, shift or production run. Data collected by SCRIBETrak is stored, analyzed, and visualized in the same manner as the automatically collected data (i.e., via the WEBTrak portal).

## **Configuring SCRIBETrak**

The SCRIBETrak data entry forms are configured via MODELTrak. Configuration options include which forms show up on which nodes and default form settings. This allows customization of the SCRIBETrak application and forms for the users designated to data entry tasks.

## **SCRIBETrak Groups and Items**

Data entry in SCRIBETrak is done through a set of forms called SCRIBETrak Items. SCRIBETrak Items can be configured and grouped into SCRIBETrak Item Groups. SCRIBETrak Groups and Items as well as their associated security permissions are configured in MODELTrak.

## **Manual Systems**

In order to allow for manual data entry, special Systems called Manual Systems must be configured in MODELTrak to receive the various events and KPI information specified. A Manual System in MODELTrak can be created by selecting the Manual Data Entry checkbox on the Advanced tab in the System properties dialog. Manual Systems are represented by a silver gear icon in the MODELTrak tree view (vs. the orange gear icon for normal "automated data collection" Systems). Systems may be changed from Manual to "automatic" at any time, however only Manual Systems are targets for manual data entry via SCRIBETrak.

## **SCRIBETrak Application**

The SCRIBETrak application is the interface for end-users to enter and edit performance data manually. SCRIBETrak can be installed on a client PC and is accessed via a Start Menu or desktop shortcut. The content available (Groups and Items as well as security and permissions) within the SCRIBETrak application is determined by the configuration in MODELTrak.

## **Main Menu**

The SCRIBETrak application consists of a list of SCRIBETrak Groups and their associated Items. The Groups and Items that show are determined by the configuration in MODELTrak.

## **Security**

When users select a specific SCRIBETrak Item, the permissions granted to that Item's Group are evaluated and access is granted or denied. If the SCRIBETrak Group is configured to allow access based on Windows Users or Groups then the current user's credentials are evaluated and they are allowed or denied access with no additional login prompts. If security is setup based on TrakSYS Users then a dialog will be displayed allowing the entry of login and password.

## **Forms**

Descriptions of the different SCRIBETrak forms can be found in the SCRIBETrak Items section of the MODELTrak section of this document.

## AUDITTrak

The AUDITTrak functionality allows for the capture of digital signatures when any change to data and/or configuration is made. This includes creating, modifying or deleting any item that is stored in the TrakSYS database.

## Configuring

The AUDITTrak capabilities may be configured using the TrakSYS Settings form available in MODELTrak (refer to the descriptions of the settings in that section of the User Guide).

AUDITTrak uses a series of database triggers to aid in tracking and recording changes to data and configuration. In addition to enabling the AUDITTrak functionality using the TrakSYS Settings, the AUDITTrak triggers must be created on the database using the **AUDITTrak | Create Triggers** menu option in the Database Manager application.

## Windows Applications

From any of the TrakSYS Windows applications (MODELTrak, LOGICTrak, SCRIBETrak and Tag/Event Definition Import, the AUDITTrak digital signature dialog is displayed as a popup form that requires one or two signatures and notes (optional). A valid username and password must be entered (based on the configured TrakSYS Settings) before the changes are committed to the database.

Only TrakSYS Users that are configured as Auditors will be allowed to sign for changes.

## WEBTrak

From WEBTrak, the AUDITTrak digital signature entry is displayed inline at the bottom of each form where data can be saved. A valid username and password must be entered (based on the configured TrakSYS Settings) before the changes are committed to the database.

Only TrakSYS Users that are configured as Auditors will be allowed to sign for changes.

## EVENTTrak

Because of the vastly different user interface requirements that may be encountered on the plant floor, the AUDITTrak digital signature dialog is implemented as a non-visual API in the EVENTTrak application. Methods of the Event Data ActiveX Control allow for username, password and notes information to be submitted when changes to event data are made. This allows a customer specific dialog to be developed with special look and feel as well as business rules for the exact situation.

The Event List exposes several new Methods to aid in the construction of an AUDITTrak dialog.

AcknowledgeWithAudit  
SaveAsNewWithAudit  
SaveWithAudit

ValidateAuditorLogin  
ValidateWindowsLogin

Only TrakSYS Users that are configured as Auditors will be allowed to sign for changes.

## Reporting

All data captured by AUDITTrak is recorded to a several tables within the TrakSYS database. The summary and details are available using the Audit Summary and Audit Dump reports available in WEBTrak.

## HISTORITrak

HISTORITrak enables TrakSYS to capture and record individual Tag value changes based on different configuration criteria. Tag values can be recorded on-change, on fixed intervals (periodic) or manually. Any type of Tag in TrakSYS can be configured for historization.

## Configuring

HISTORITrak is configured at the Tag level in MODELTrak. The HISTORITrak tab on the Tag Properties dialog allows for the creation of one or more Tag History Definitions. These items specify how and when Tag value changes are recorded to the database.

## Data Capture and Expiration

Tag values are monitored and recorded based on the configuration by the TrakSYS LOGICTrak Service. If the Expiration Hours property of the Tag History Definition is set to a value greater than 0, the TrakSYS Maintenance Service will constantly clear Tag history data outside the range for each Tag.

## Manual Capture

In some cases, it is desirable to capture Tag values when certain conditions are present (rather than on-change or periodically). Using the Advanced Script Tag API *Global.Context.CreateTagHistoryRecord*, values can be saved for a specific Tag when user defined logic and conditions are present.

## Reporting

All Tag data captured by HISTORITrak is recorded to a several tables within the TrakSYS database. Several HISTORITrak Web Parts are available in the dashboard reporting to display Tag history trends and comparisons with other KPIs.

## IMPROVETrak

The IMPROVETrak Journal allows for manual entry of noteworthy improvement related events (start of initiatives, updates, setpoint changes, end of shift notes, etc...).

## Configuring

The only pre-configuration for IMPROVETrak is to optionally define Journal Categories in MODELTrak. Journal Categories may be applied to entries for filtering and grouping in the IMPROVETrak reports.

## Data Entry

IMPROVETrak Journal entries are made using a web based editor form in WEBTrak (see the New Journal Entry Report Type).

## Reporting

All entries made into IMPROVETrak are recorded to tables within the TrakSYS database. The summary and details are available using the Journal Summary reports available in WEBTrak.

## SPCTrak

SPCTrak allows TrakSYS to capture and report product sample measurements and other process variables for statistical analysis. Random samples can be entered manually via EVENTTrak or captured automatically using APIs in LOGICTrak. Standard SPC reports are available in WEBTrak.

## Configuring

The main configuration point for SPC data is called a Sample Definition. Systems may be configured to capture one or more Sample Definitions. These settings can be made in the SPC tab in the System Properties dialog in MODELTrak. Once Sample Definitions are defined, data can be gathered manually through forms in EVENTTrak or automatically by LOGICTrak.

## Data Entry from EVENTTrak

Because of the vastly different user interface requirements that may be encountered on the plant floor, the SPC data entry is implemented as a non-visual API in the EVENTTrak application. The Sample Sub-Group Data Control Activex allows for sub-groups and samples to be saved to the SPCTrak data tables. This allows a customer specific forms to be developed with special look and feel as well as business rules for the exact situation.

## Automatic Data Capture

In some cases, it is desirable to capture SPC data automatically when certain conditions are present. Using the Advanced Script Tag API *Global.Context.CreateSampleSubGroup*, values can be saved for a specific Sample Definition when user defined logic and conditions are present.

## Reporting

The SPC sub-group and sample data captured by SPCTrak is recorded to a several tables within the TrakSYS database. Standard SPC control chart Web Parts are available in the dashboard reporting to display SPCTrak trends for analysis.

## BATCHTrak

BATCHTrack is a collection of features that allows TrakSYS to model and record data from Batch production environments. The following sections describe the basic features and options of BATCHTrak. Detailed explanations of the BATCHTrak entities can be found in the MODELTrak section of this document.

## Configuration

Configuring BATCHTrak is comprised of the following basic steps.

### Systems / Function Definitions

Batch Systems are defined in the Areas and Systems section of MODELTrak. Each Batch System must contain at least one Sub-System. Function Definitions are then defined under the Sub-Systems. A Function Definition represents an operation that the Batch System may perform. The actual sequence and duration of the Function Definitions needed to create a Batch is defined later in the Batch Recipe section.

### Materials

Optionally, Materials can be defined for use in Batch Recipes. The Materials node (in the PRODUCTTrak section of MODELTrak) allows raw Materials as well as compound Materials (several other Materials combining to form a new Material) to be defined.

### Product / Recipe

Before creating Batch Recipes, Products must be defined. If Materials are being used, the main Material being consumed is assigned to the Product. Last, a Recipe must be created which defines the steps (Function Definition sequence and durations) that required to produce each Batch for a specified Product.

## Reporting

BATCHTrak reporting is achieved using a combination of WEBTrak dashboard web parts and direct access to the Job/Batch data tables in the TrakSYS database. Refer to the online TrakSYS Knowledgebase for report samples and database schema documentation.

## Database Manager

Database Manager is a utility application which allows you to configure the settings that MODELTrak, LOGICTrak, and WEBTrak Knowledge Management Portal use when connecting to the TrakSYS database. It also includes options for creating new databases, upgrading databases from previous versions, repairing the default database logins, and synchronizing the report data source.

The section below describes the Database Connection Settings that will be used by all of the applications to connect to the database:

General	
Database Server	Name of the PC hosting SQL Server and the database.
Database Name	Name of the TrakSYS database. The default name is EDB.
Security Type	Whether to use SQL Server Authentication or Windows Authentication to connect to the database. By default,

	TrakSYS uses SQL Server Authentication.
Database Login	The login or user name to use for connecting to the database. The default is edbApp.
Database Password	The password corresponding to the specified Database Login. This setting is encrypted for security.

<b>Advanced</b>	
Connect Timeout (Seconds)	The delay before a database connection attempt will fail. Larger values give applications more time to attempt connections, but may sacrifice responsiveness if the database server is inaccessible or unresponsive. The default setting is 30 seconds.
Template	The actual connection string template applications use to establish database connections. Use this setting to add or modify connection options which Database Manager does not directly expose. Note that connection strings should always use the SQL Server provider.

## New Database

This option launches the EDB setup, which allows you to create a new TrakSYS database on the SQL Server of your choice and gives you the option of installing sample data.

## Upgrade Database

This option activates the database upgrade wizard to convert a TrakSYS database from an older version to the current schema. See Upgrading From a Previous Version for more details.

## Repair Logins

This option repairs the default edbApp, edbAdmin, and edbUser logins damaged by restoring the TrakSYS database backup made from a different PC than restored to. Use this to repair the logins any time they cease to function after restoring a database. The login specified for this operation must have sufficient administrative rights to create and delete database logins, users and roles. It is recommended that the SQL Server "sa" login is used for the Repair Login operation.

## Check Referential Integrity

This menu option scans the entire TrakSYS database for referential integrity data inconsistencies. Under normal operations, triggers and other database techniques are employed to prevent this from occurring. This menu option is typically only required for technical support use.

## Archive Database

Data archiving is achieved by allowing the entire EDB database to be backed-up and restored. This allows for the archive data to be easily moved off the data collection server. The archive process copies all of the configuration and data into the specified



archive database. The archive database is altered and cannot be used as a TrakSYS data source.

## **Archive Database Wizard**

The Archive Database Wizard provides options for specifying the database to be archived.

### **Database Connection Settings**

The Database Connection Settings dialog allows for the connection to be specified for the source database for archiving. The login specified must have sufficient rights for database creation and deletion on the target database server.

### **Archive Database Settings**

The Archive Database Settings dialog allows for advanced settings to be modified that affect how the archive process is executed.

The Archive Database Name will be the name of the new archive database. Its default value is the name of the source database with "\_Archive" appended.

The Temporary Backup File is the path and filename for the backup file that is created as part of the archive process. This file will be deleted after the archive operation is complete. The location specified should have sufficient disk space for the EDB database during the archive process.

The Archive Database File Path is the target for the data and log files that will be created for the new archive database. By default, the Archive Database File Path is automatically calculated during the archive process. It is typically the default data and log file location for the SQL Server.

## **Configuring Reports for Archive Data**

Archived data can be accessed by configuring a new set of Reporting Services reports for the Archive database and configuring WEBTrak reports with a custom Connection String. The following sections describe this process.

### **Creating a Duplicate Report Set**

The TrakSYS reports are located in a folder called "TrakSYS Reports" in the Reporting Services web. All of the reports use a data source called EDB located in that folder. In order to configure reports to access an Archive database, a duplicate set of reports must be created with its own EDB data source. The following steps describe creating a duplicate set of reports in SQL Reporting Services.

1. Open the SQL Reporting Services web (<http://<servername>/reports>).
2. Rename the "TrakSYS Reports" folder to "TrakSYS Archive Reports".
3. Open the EDB data source under "TrakSYS Archive Reports" and change the connection string properties to reference the Archive database.
4. Open the TrakSYS Database Manager and select the Reports | Upgrade/Publish Report Files option to recreate the main set of reports ("TrakSYS Reports").

## Changing WEBTrak Report Connection String

Each report in WEBTrak contains a Connection String parameter which determines the database that is used to retrieve parameter selection values (such as Area, System, Shifts, etc...). By default, the Connection String parameter is empty indicating that the TrakSYS master application connection string is used. To configure reports to retrieve report parameter selections from the Archive database, change the Connection String parameter to a valid SQL Server connection string to the Archive database. A sample connection string format is provided below. The tokens in {} should be replaced with the appropriate values. {integratedSecurity} should be set to False when a SQL Server login and password is provided and True for Windows Integrated Security.

```
Data Source={databaseServer};Initial Catalog={databaseName};User
ID={databaseLogin};Password={databasePassword};Integrated
Security={integratedSecurity};Connect Timeout={connectTimeout};
```

## Changing the WEBTrak Report Target URL

Reports URLs must be edited in Design View to reference the "TrakSYS Archive Reports" folder. By default report URLs are configured using the {reportRootPath} Report Variable which has a value of "/reportserver?/TrakSYS Reports/". Replace the {reportRootPath} variable with the constant value of "/reportserver?/TrakSYS Archive Reports/" in order to access the Archive report set.

## Delete Data

The Database Manager application provides data clearing capabilities to remove ranges of collected event and OEE data from the database. The Delete Data Wizard can be launched from the Database | Delete Data menu item.

### Delete Data Wizard

The Delete Data Wizard provides options for specifying the data to be removed.

### Database Connection Settings

The Database Connection Settings dialog allows for the connection to be specified for the database where data should be deleted. The login specified must have sufficient rights for data deletion in the target database.

### Delete Data Settings

The Delete Data Settings dialog allows for the date range to be specified for data removal. All event and OEE data between the From Date and To Date (inclusive) will be permanently deleted from the database. The data range that is available in the database is expressed by the Oldest and Most Current Data labels.

## Update Data Source

The Reporting Services Data Source is a separately stored set of database connection credentials used by the Reports to connect to the database. Whenever the application database connection settings are changed via the Database Manager, the Reporting Services Data Source is automatically updated. The automatic update feature can be turned on and off by toggling the value of the Reports | Auto Update Data Source menu item. The Reporting Services Data Source can be manually updated by selecting the Reports | Update Data Source item.

The automatic update feature assumes that the Reporting Services is installed with default settings on the same server as the EDB database is located. If the Reporting Services installation is on a different server or the Reporting Services webs have been renamed from their defaults, then the Data Source must be updated manually using the Reporting Services web interface.

## **Upgrade/Publish Report Files**

This menu option publishes the standard TrakSYS report files to the MS SQL Server Reporting Services site. This option may be used to restore the original report files and may also be used to update report files during patching or Service Packs.

## **Create Audit Triggers**

This menu option is used to create the database triggers necessary for the AUDITTrak feature to function. Triggers are created on each database table that aid in capturing changes, inserts and deletes.

## **Delete Audit Triggers**

This menu option is used to remove the AUDITTrak database triggers. This should be run if the AUDITTrak feature is being disabled or removed as the AUDITTrak triggers cause extra overhead to take place on database actions.

## **Create Notification Instance**

This menu option creates the necessary SQL Server databases to support the ALERTTrak messaging and notification capabilities. SQL Server 2005 is required for ALERTTrak.

## **Delete Notification Instance**

This menu option clears the SQL Server databases and settings created to support the ALERTTrak messaging and notification capabilities. This option is intended as a troubleshooting / reset technique when problems with the notification databases are occurring. This can also be used before a TrakSYS uninstall as the notification databases are not cleared as part of the standard uninstall procedure.

## **Resync Notification Database**

This menu option syncs the ALERTTrak notification databases with the configuration information contained in MODELTrak and the main TrakSYS database (usually EDB). There are several pieces of configuration that are duplicated in the main TrakSYS database and the notification database. During normal operations, the configuration in these databases are maintained and synchronized without the need for manual interaction. However, during development or recovery scenarios where databases are backed up and restored, it is possible for the data in the notification databases to become out of sync with the TrakSYS database. This function clears the notification databases and synchronizes them with the required configuration from the main TrakSYS database.

## Register Remote Notification Instance

This menu option creates the necessary registry entries required for the TrakSYS LOGICTrak Service to interact with the ALERTTrak notification databases. This procedure is done automatically when the Notification instance is created so it is not necessary to run this when LOGICTrak is running on the same server as SQL Server. This option is only required when running LOGICTrak on a different computer than the SQL Server.

## License Manager

The License Manager application manages all aspects of TrakSYS licensing. It allows you to install new license features, view features already installed, activate features that require activation, and delete expired or unwanted license features.

After completing the TrakSYS setup you must install the license file supplied by your distributor before the software will function. To do this, select Import License File from the Feature menu to browse for your XML license file. License Manager will import all valid features in the file and display them in the main list view.

You only need to install your license once. The License Manager stores installed license features centrally in the TrakSYS database. There is no need to install licenses on every computer, such as EVENTTrak nodes or WEBTrak Knowledge Management Portal clients. Remote components verify licensing upon connecting to the central database.

Certain feature lines require host names in order to function; these include MODELTrak, LOGICTrak, and WEBTrak. You will not be able to use these features until you specify the names of the computers which host these components.

The main Features list view displays all installed license features. Additional details for an individual feature are available by right-clicking on a feature and selecting Details or by selecting Details from the Feature menu. The details dialog displays the following information:

<b>General</b>	
Name	The name of the selected feature, which may correspond to a particular application or one of several licensed Configuration Entities.
Version	The version of TrakSYS for this license feature. This value must match the version of TrakSYS you are using or the feature will have no effect.
Key	The unique alphanumeric key identifying and verifying the authenticity of this feature.
Notes	Vendor supplied notes for this feature.
Data	Feature specific data, which is typically an integer indicating the maximum number allowed. For example, if your Event Definitions feature's data value is 8000, you are allowed a maximum of 8000 Event Definitions. In some cases a value of * or 1 indicates that the feature is enabled.
Host	This feature's host name, which limits the feature to a particular computer. Some features may limit you to one

	host at a time. You must designate specific host names for the MODELTrak, LOGICTrak, and WEBTrak feature lines.
<b>Numbers</b>	
Serial Number	The serial number identifying a particular license. All features in a license file have the same serial number.
Item Number	The item number identifying a particular feature within a license. These are usually sequential within a license file.
Part Number	Part number reserved for internal use which identifies the feature.
<b>Customer</b>	
Customer Name	The customer who purchased the license, which is typically the name of the purchasing company.
Customer Site	The intended site of the license, used for recordkeeping purposes.
Customer Address	The address of the purchasing customer.
<b>Expiration</b>	
Expiration Date	A specific date that this feature will expire. When features expire, you must obtain another license in order for the software to continue functioning.
Expiration Days	If present, the number of days this feature will last before expiring. When this value is nonzero, the countdown begins once you begin recording data.
Expiration Days Remaining	The calculated number of days remaining until this feature expires. This is based on the minimum number of days based on the Expiration Date and Expiration Days attributes.
<b>Activation</b>	
Activation Required	Indicates whether or not this feature requires activation. Refer to the section on Activation for more information.
Activation Key	If present, the activation key provided when you activated the software. Refer to the section on Activation for more information.

A separate Usage tab, located beside the Features tab, displays the number of Event Definitions, KPI Calculations, and EVENTTrak Nodes your configuration uses relative to the limits of your license. This information may help you anticipate the need for a larger license before you exhaust the limits of your current license.

## File License Mode

This menu option toggles the File License Mode option which allows TrakSYS to look for the license information in a file rather than the standard database location. A file based license is typically used in a development situation where a single server or PC is used against multiple databases regularly.

## Activation

Some license files require activation before they will function. Such features will clearly indicate that you must activate them. Activation validates your license with the software vendor to prevent other parties from attempting to use your license.

License Manager supports both automatic online activation and manual offline activation. Before activating, be sure to set the Host setting for each applicable

feature line. Changing the Host values later will require you to reactivate your license.

To begin automatic online activation, select Activate Online from the Activation menu. License Manager will securely connect to the online activation site using the active Internet connection. The operation may fail if firewall or other security settings interfere.

To activate your license manually, begin by selecting Export Activation Data from the Activation menu to generate an activation Cypher file. Once you submit this to your software vendor, they will provide you with an activation Key file. Select Import Activation Data from the Activation menu to import the activation Key file and complete the activation process. When activated, feature states will display VALID in the main list view and their activation Keys will be visible in the details view.

## Tag Import

The Tag Import application facilitates batch importing Tags into the EDB database using a special CSV file populated with Tag data. Use the Tag Import application to select a CSV file and import its Tags. Advanced options available in the application allow for customization of the expected format of the CSV import file and the default import behavior.

### Tag Import Format

The Tag Import application imports special text CSV files. A Microsoft Excel Template file demonstrates the exact format of import files, and includes embedded comments which explain the details of what data must be in each column.

After filling out the Excel Template, save the resulting CSV file by selecting the File | Save As menu option and selecting the CSV file format. After saving the CSV file, close Excel before using the Tag Import application.

**IMPORTANT!** The last column of the Excel Template (Store) must always contain a 1 or a 0 for every row. This ensures that the exported CSV file contains a value for every column.

### Tag Import Wizard

Use the Tag Import wizard to select a CSV file to import. The first screen of the wizard selects the file and optionally specifies advanced options for controlling the expected CSV format and import behavior. The following table explains the Advanced Settings.

Log File	This is the full path and file name for the results log file that will be created for the Tag Import. If this field is left blank, the Log File will be created in its default location which is the same directory as the selected CSV import file.
Delimiter	This character will be interpreted as the column delimiter. The default value is a comma (,).
Comment	This character can be used in the CSV text file to indicate a comment line which will be ignored by the import. The comment character must be the first character on the line. The default value is (#).

Quote	This character is used in the CSV file to surround column values. Its use is optional and is only necessary when the column value includes the Delimiter character. The default value is (").
Escape	This character is used to escape the Quote character if it is part of a column value. The default value is ("). For example, if the column value is "This is "the" value", the escape character must be used to make the value "This is ""the"" value".
Has Headers	This setting indicates if the first row of the CSV is a header row or a row of column values.
Trim Fields	This setting determines if extra white space at the beginning and end of column values is automatically trimmed during import.
Overwrite Records	This setting determines the import behavior when Tags that are being imported already exist in the database.

## Event Definition Import

The Event Definition Import application facilitates batch importing Event Definitions into the EDB database using a special CSV file populated with Event Definition data. Use the Event Definition Import application to select a CSV file and import its Event Definitions. Advanced options available in the application allow for customization of the expected format of the CSV import file and the default import behavior.

## Event Definition Import Format

The Event Definition Import application imports special text CSV files. A Microsoft Excel Template file demonstrates the exact format of import files, and includes embedded comments which explain the details of what data must be in each column.

After filling out the Excel Template, save the resulting CSV file by selecting the File | Save As menu option and selecting the CSV file format. After saving the CSV file, close Excel before using the Event Definition Import application.

**IMPORTANT!** The last column of the Excel Template (Duration) must always contain a value in every row. This ensures that the exported CSV file contains a value for every column.

## Event Definition Import Wizard

Use the Event Definition Import wizard to select the CSV file for importing. The first screen of the wizard selects the file and optionally specifies advanced options for controlling the expected CSV format and import behavior. The following table explains the Advanced Settings.

Log File	This is the full path and file name for the results log file that will be created for the Event Definition Import. If this field is left blank, the Log File will be created in its default location which is the same directory as the selected CSV import file.
----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Delimiter	This character will be interpreted as the column delimiter. The default value is a comma (,).
Comment	This character can be used in the CSV text file to indicate a comment line which will be ignored by the import. The comment character must be the first character on the line. The default value is (#).
Quote	This character is used in the CSV file to surround column values. Its use is optional and is only necessary when the column value includes the Delimiter character. The default value is (").
Escape	This character is used to escape the Quote character if it is part of a column value. The default value is ("). For example, if the column value is "This is "the" value", the escape character must be used to make the value "This is ""the"" value".
Has Headers	This setting indicates if the first row of the CSV is a header row or a row of column values.
Trim Fields	This setting determines if extra white space at the beginning and end of column values is automatically trimmed during import.
Overwrite Records	This setting determines the import behavior when Event Definitions that are being imported already exist in the database.

## Efficiency Metrics in TrakSYS

TrakSYS offers a variety of metrics for evaluating your efficiency. These include OEE, TEEP, and Takt Time.

### OEE

OEE (Overall Equipment Effectiveness) is an industry standard calculation for measuring efficiency. It is often a key metric in Total Productive Maintenance (TPM) programs, which improve the overall effectiveness and efficiency of manufacturing plants by creating a joint responsibility between operators and maintenance personnel to achieve zero product defects, zero mechanical breakdowns, and greatly reduced changeover times.

OEE makes it possible to consistently measure the effectiveness of TPM programs and other productivity improvement initiatives.

### Calculating OEE

TrakSYS uses the following standard equations to calculate OEE:

$$\text{OEE} = \text{Availability} \cdot \text{Performance} \cdot \text{Quality}$$

$$\text{Scheduled Time} = \text{Total Time} - \text{System Not Scheduled Time}$$

Scheduled Time relates to your Schedules. Gaps in your Schedule and Schedule Pattern Items with the Scheduled setting unchecked represent unscheduled time (System Not Scheduled Time), and do not impact Availability.



Production Time = Total Time – System Not Scheduled Time – Availability Loss Time

Production Time is the total time the production line is actually running. It excludes all periods of unscheduled time (System Not Scheduled) and Availability Loss events. Availability Loss is the total time that Event Definitions with the OEE Event Type setting of Availability Loss are active.

$$\text{Availability} = \frac{\text{Production Time}}{\text{Scheduled Time}} = \frac{\text{Scheduled Time} - \text{Availability Loss Time}}{\text{Scheduled Time}}$$

Availability represents the percentage of time your process is available for production.

$$\text{Performance} = \frac{\text{Total Calculation Units Count}}{\text{Theoretical Rate} \cdot \text{Production Time}}$$

Performance represents the ratio of your actual production speed versus the theoretical production rate. The Theoretical Rate (Calculation Units per Second) is part of the KPI Calculation.

$$\text{Performance} = \frac{\text{Production Time} - \text{Performance Loss Downtime}}{\text{Production Time}}$$

An alternative way to calculate Performance uses time. Assuming your production line runs at 100% of its theoretical rate and produces zero units during Performance Loss events, you can calculate Performance as the ratio of non-Performance Loss time versus Production Time. This method is typically less accurate than the standard Performance calculation, but is useful when there are no automated counters available on the production line. TrakSYS does not use this method of calculating Performance.

$$\text{Quality} = \frac{\text{Good Calculation Units Count}}{\text{Total Calculation Units Count}}$$

Quality is the percentage of total units produced that are good. Total, Good, and Bad units produced are related as shown below:

Total Calculation Units Count = Good Calculation Units Count + Bad Calculation Units Count

Refer to the section on KPI Units for more information on TrakSYS's OEE unit conversions.

## **TrakSYS OEE Implementation**

The LOGICTrak records OEE data for a KPI Calculation into discrete periods of time called intervals. Each interval contains the number of Total, Good, and Bad units LOGICTrak counted during its duration, converted to Calculation Units. The section on KPI Units explains these units and their conversions in more detail.

LOGICTrak ends intervals according to the Interval Duration setting of the KPI Calculation, but may split them sooner if the current Shift changes, a KPI Capture Tag configured to split the interval changes, or the current Product changes. Regardless, LOGICTrak always synchronizes intervals with the KPI Calculation's Interval Midnight Offset setting.

The Interval Midnight Offset setting is an offset from 12:00 AM to use as a user-definable midnight for synchronizing KPI Intervals. For example, if you set the Interval Midnight Offset to +30 minutes, and then set Interval Duration to 60 minutes, LOGICTrak will write KPI Intervals as 12:30 AM, 1:30 AM, 2:30 AM, and etc, even if Capture Tag, Shift, or Product changes force Interval splits between those times. A common use of this setting is to synchronize KPI Intervals with the Shift Schedule.

LOGICTrak does not explicitly calculate OEE; it merely records the data necessary for the OEE reports. Other reports use the same data to calculate different metrics such as TEEP and Takt Time. The database does not contain any calculated performance data such as Availability, Performance, Quality, or OEE.

## TEEP

TEEP (Total Effective Equipment Performance) is a variation of OEE that reinterprets unscheduled time as Availability Loss. It represents the percentage of good performance relative to calendar time, uncorrected for unscheduled time, and better indicates your potential for improvement. TEEP is always lower than OEE; any non-production time, even scheduled maintenance and holidays, will lower your TEEP. It is analogous to scheduling your lines to run 24 hours a day and 7 days a week.

$$\text{TEEP Availability} = \frac{\text{Production Time}}{\text{Total Time}} = \frac{\text{Total Time} - \text{Availability Loss}}{\text{Total Time}}$$

Availability is the only factor of TEEP calculated differently than OEE. Note that TEEP Production Time does not exclude System Not Scheduled Time. The remaining factors, Performance and Quality, are identical to those in OEE.

$$\text{TEEP} = \text{TEEP Availability} \cdot \text{Performance} \cdot \text{Quality}$$

## Takt Time

Takt Time is the amount of time required to produce one good unit. It is especially useful for matching production to customer demand.

$$\text{Takt Time} = \frac{\text{Production Time}}{\text{Good Calculation Units Count}}$$

Takt Time relates to OEE Performance and Quality as shown below:

$$\text{Takt Time} \cdot \text{Theoretical Rate} = \frac{1}{\text{Performance} \cdot \text{Quality}}$$

KPI Calculations allow specification of Target Takt Time as Theoretical Rate (Seconds per Unit) settings. This appears as a target line relative to the calculated Takt Time in the Report graphs.

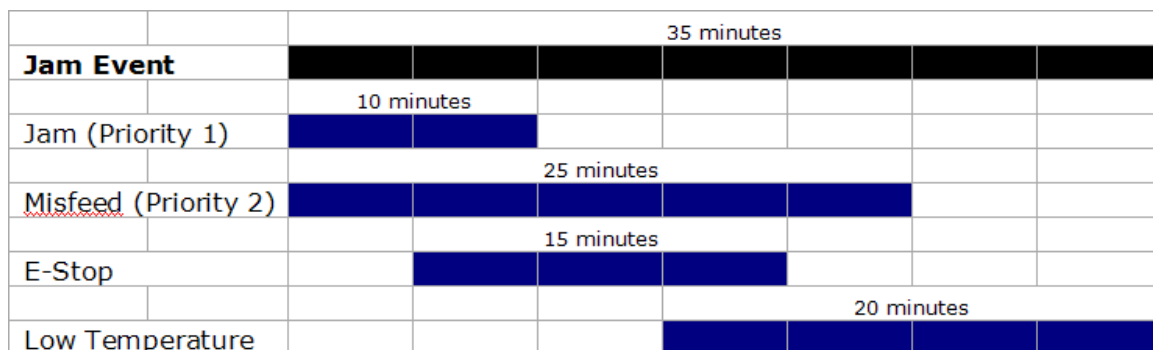
## Advanced Topics

This section covers advanced topics that may not be necessary for everyday use of TrakSYS, but will help you better understand the data and troubleshoot less common problems.

### Sub-Events

LOGICTrak records active Event Definitions as Events and Sub-Events. An Event represents a contiguous period of time one or more of a System's Event Definitions are active. When two or more Event Definitions become active at the same time, LOGICTrak assigns the Event Definition with the lowest numerical priority as the root cause of the Event. It logs the remaining active Event Definitions (including the root cause) as Sub-Events.

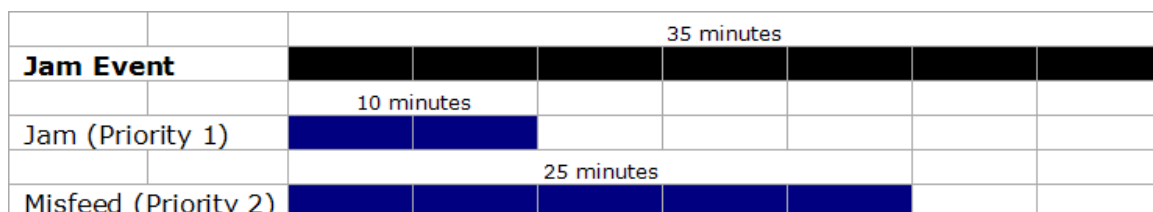
The example below depicts a sample Event and its Sub-Events for a hypothetical System with Jam, Misfeed, E-Stop, and Low Temperature Event Definitions.



In the chart above, the Jam Event Definition is the root cause because it has a Priority of 1, whereas the Misfeed Event Definition has a Priority of 2. The Misfeed, E-Stop, and Low Temperature Sub-Events are only visible in a detail report. For reporting purposes, the Jam Event is responsible for the entire 35 minute System stoppage.

### Event Definition Priority

When two or more Event Definitions become active at the same time, or, in the case of splitting, are already active, LOGICTrak uses Priority to determine which Event Definition is the root cause of the Event. LOGICTrak always assigns responsibility to the active Event Definition with the highest priority (lowest numerical value), while it records the remaining active Event Definitions as Sub-Events.



In the example above, the Jam becomes the root cause because it has a higher Priority than Misfeed. Note that Priority only has an effect when two or more Event Definitions become active within the same LOGICTrak scan. If an Event Definition with lower priority (higher numerical value) becomes active before an Event Definition with higher priority (lower numerical value), then the first, lower priority Event Definition will be the root cause of the Event regardless of its priority.

To force higher priority Event Definitions to take precedence over lower priority Event Definitions already in progress, use the "Re-Evaluate System Event on Start" setting to force LOGICTrak to split Events when higher priority Event Definitions become active. Note that this will not cause the higher priority Event Definition to take responsibility for the current Event; instead, LOGICTrak will end the current System Event and immediately start a new System Event using the higher priority Event Definition as the root cause.

## Sub-Priority

In some cases, Priority alone is not always sufficient to produce desired Event splitting. To address these rare cases, TrakSYS offers Sub-Priority. Consider the following two sample System Events:

		25 minutes			
<b>Jam Event</b>					
		10 minutes			
Jam (Priority 1.1)					
		20 minutes			
Misfeed (Priority 1.2)					

In the example above, the Misfeed Event Definition should not split the Event even if it has "Re-Evaluate System Event on Start" option set because it has the same Priority (1) as Jam, which started first. Note that Jam and Misfeed's Sub-Priorities, 1 and 2 respectively, have no effect. You could accomplish the same behavior by assigning Misfeed a lower priority to ensure it never splits a higher priority Jam Event, but then a Jam would always split a Misfeed Event, making the following example impossible:

		25 minutes			
<b>Misfeed Event</b>					
		20 minutes			
Jam (Priority 1.1)					
		25 minutes			
Misfeed (Priority 1.2)					

When Jam and Misfeed have the same Priority, the first one to begin always remains the root cause of the System Event. This is typically the desired behavior for higher priority Event Definitions. However, when two higher priority Event Definitions begin at the same time, it is ambiguous which will become the root cause of the new Event. Consider this example:

		<b>Inbalance Event</b>	<b>Jam Event</b>
		10 minutes	15 minutes
		10 minutes	15 minutes
<u>Inbalance</u> (Priority 2)			
			15 minutes
Jam (Priority 1.1)			
			15 minutes
<u>Misfeed</u> (Priority 1.2)			

In the example above, it is unclear whether the new Event's root cause will be the Jam or the Misfeed without Sub-Priority. To ensure the previous two examples remain possible, both Jam and Misfeed must have the same Priority. Therefore, in this scenario, you must use Sub-Priority to specify whether the Jam or the Misfeed should become the root cause of the new Event. As indicated above, Jam will become the root cause in this example because it has a higher Sub-Priority (1) than Misfeed (2). The remaining active Event Definitions, Inbalance and Misfeed, will become Sub-Events, which are only visible in a detail report.

## Sub-Systems

Sub-Systems provide an additional level of grouping within normal Systems. This is useful for configuring Overview Systems; in these cases, Systems typically represent entire production lines while Sub-Systems represent equipment. With such a hierarchy you can easily relate line stoppages to the equipment and Event Definitions that caused them. The reports allow you to group your data by Sub-System, providing a clear view of the root cause for each interruption.

With or without Sub-Systems, a System can only have one active Event at a time. If multiple Sub-Systems have active Events Definitions at the same time, LOGICTrak will relegate these to Sub-Events. Therefore, using Sub-Systems sacrifices some accessibility of detailed equipment stoppage information in favor of more hierarchical aggregate data. Secondary Events not directly responsible for line stoppages, such as faults in other equipment occurring during ongoing System Events, will only be viewable in a detail report.

## Filtering Erratic KPI Counter Behavior

TrakSYS provides several advanced settings to filter erratic KPI Counter behavior and prevent corrupt efficiency data that may prove difficult to troubleshoot. Ideally, Counter Tags should never move backward or unexpectedly jump forward in value, but several common scenarios, such as power loss, uploading new programs to PLCs, or I/O communication interruptions, can cause such undesired behavior.

To remedy these situations, KPI Counters include a "Maximum Increment per Scan" setting which limits forward Counter movement to the specified value. LOGICTrak immediately discards any Counter movement exceeding this setting, preventing sudden jumps in Counter values from skewing OEE data. Configure this setting according to the maximum possible amount the Counter can increment within one LOGICTrak scan.

As a rule, LOGICTrak never allows backward Counter movement. Any time a Counter moves backward, LOGICTrak interprets it as Rollover, subject to the "Maximum Increment per Scan" setting for the KPI Counter. LOGICTrak uses the following equation to interpret backward counter movement:

$$\Delta_{\text{counter}} = \begin{cases} (\text{NewValue} - 0) + (\text{Rollover} - \text{LastValue}) & \text{if Rollover} > 0 \\ (\text{NewValue} - 0) & \text{if Rollover} \leq 0 \end{cases}$$

In the equation above, if the change in Counter value exceeds the "Maximum Increment per Scan" setting, it is effectively 0 for that scan. LOGICTrak writes messages to the database log any time Rollover occurs or the Counter jumps forward by more than the maximum, allowing you to easily diagnose erratic Counter behavior and adjust your KPI Counter settings appropriately. The Log Summary report in WEBTrak allows you to easily view these log entries.

## KPI Counter Flow

TrakSYS uses "Flow" to indicate when a line is running, or more specifically, when product is flowing through a KPI Counter or Input. One possible use of Flow is to automatically end a manual Event such as Lunch once the line resumes running.

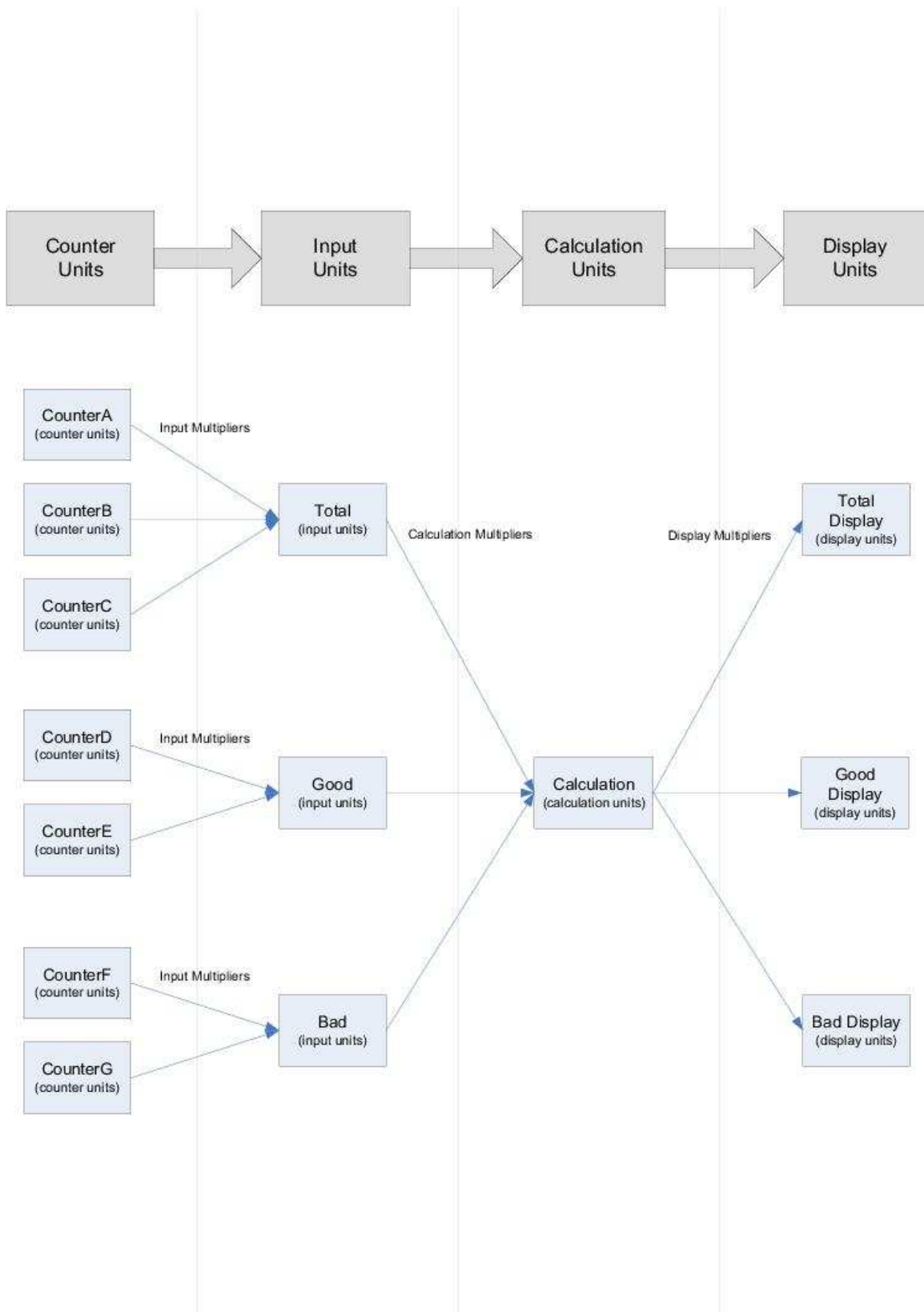
MODELTrak provides three basic options for adjusting each type of Flow. The first two, Flow On Count and Flow On Period, specify when the Flow should become active. Flow On Count specifies how many units must pass through the Counter (or Input) within the Flow On Period before the Flow activates. For example, if the Flow On Count is 5 and the Flow On Period is 7 seconds, then Flow will activate when the Counter (or Input) increments by 5 units in 7 seconds or less. Note that the Flow On Period is not a delay; if 5 units pass in 2 seconds, the Flow will activate in 2 seconds.

The remaining setting, Flow Off Delay, specifies a delay after units stop flowing before Flow deactivates. For example, if the Flow Off Delay is 15 seconds, Flow will not stop until 15 consecutive seconds elapse without a single unit passing.

KPI Counters and OEE Inputs expose Flow information via State Tags, accessible as the "Flowing" and "FlowDuration" attributes. Note that in order to access Flow information for an OEE Input, you must create a State Tag, select "KPI Calculation" as the Entity Type, select the desired KPI Calculation as the Entity, and then select the Flowing or FlowDuration attribute of the Total, Good, or Bad OEE Input; OEE Input is not directly available as an Entity Type because there are three for every KPI Calculation.

## KPI Units

LOGICTrak and the reports must convert raw KPI Counter values into consistent sets of units in order to calculate KPI. TrakSYS defines the following sets of units and their conversions:



## KPI Units and their Conversions

## Counter Units

Counter Units are the actual values provided by I/O Counter Tags. Since an Input (i.e. Total, Good, or Bad) may be a combination of several I/O Tag values, KPI Counters within the same Input might use different Units depending on their placement on the production line. TrakSYS uses the "Counter Units to Input Units Multiplier" setting of a KPI Counter to convert its values into consistent KPI Input Units; however, this setting is usually unnecessary because counters within an Input usually have the same units.

## Input Units

Input Units represent a consistent set of units within a single Input (i.e. Total, Good, or Bad). TrakSYS uses the "Counter Units to Input Units Multiplier" setting to convert KPI Counter values to Input Units.

If all of the Counters for a particular Input share the same units, it may be preferable to set the "Input Units to Calculation Units Multiplier" setting for the entire Input rather than setting the "Counter Units to Input Units Multiplier" for each Counter. For example, if all of your Counters for the Total Input count boxes but your KPI Calculation uses wallets, set the appropriate multiplier on the Input rather than each Counter.

## Calculation Units

Calculation Units are a consistent set of Units for an entire KPI Calculation. TrakSYS converts all KPI Counter and Input values to Calculation Units in order to calculate OEE. These units typically correspond to the smallest Counter Units on the line, which are often the Total Counter's units.

## Display Units

The reports use Display Units to display Total, Good, and Bad unit counts. Every KPI Input has its own "Display Units" label (i.e. boxes, cartons, or tablets) and "Calculation Units to Display Units Multiplier" setting.

## OPC Write-Back

TrakSYS provides the option for writing values back to process control devices and HMIs using the OPC communication layer. Throughout the product, opportunities to write values to Tag locations now include the ability for those target Tags to be specified as OPC (as opposed to exclusively Virtual Tags in the past). This includes assigning values using Assign Tag logic, Product Mapping, Batch Parameter assignment, and writes using the API in Script Tags and classes.

## OPC Troubleshooting Guide

This guide describes a variety of strategies for troubleshooting common OPC problems. OPC relies on DCOM, which is a complex protocol subject to many different security and configuration settings. Refer to your OPC I/O server's documentation if these guidelines do not solve your problem.

## TrakSYS Setup OPC Preparation

The TrakSYS setup attempts to prepare servers for OPC communication using the following procedures:



- Adds MODELTrak and the LOGICTrak Service to the Windows Firewall exceptions list when installing on Windows XP with Service Pack 2 or higher.
- Opens TCP port 135 in Windows Firewall when installing MODELTrak or LOGICTrak Service on Windows XP with Service Pack 2 or higher.
- Ensures that DCOM is enabled in the registry using the following settings in the HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Ole key any time you install MODELTrak, LOGICTrak Client, or LOGICTrak Service:

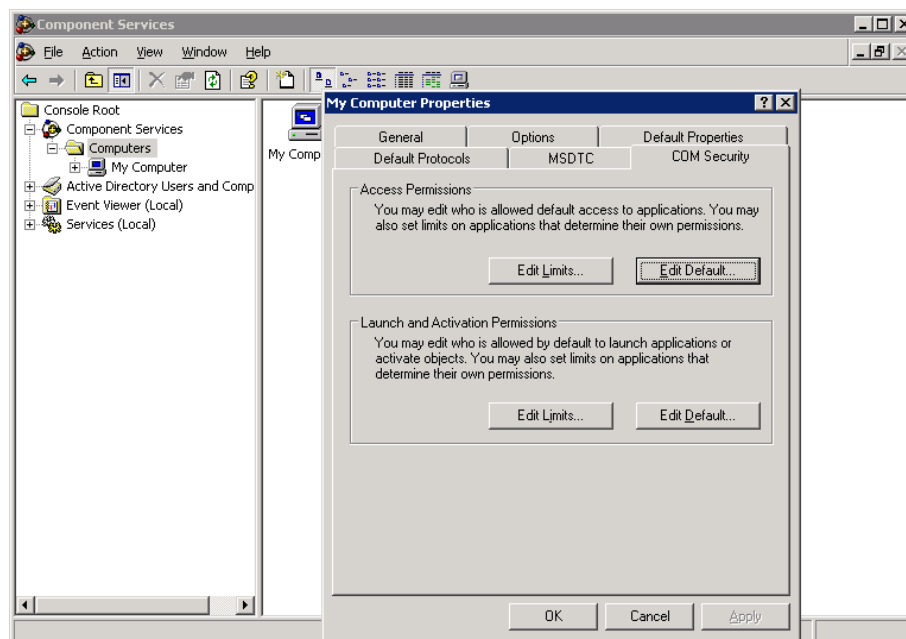
Value Name	Data Type	Value
EnableDCOM	String	Y
EnableDCOMHTTP	String	N
EnableRemoteConnect	String	Y
LegacyAuthenticationLevel	DWORD	2
LegacyImpersonationLevel	DWORD	2
LegacySecureReferences	String	N

The TrakSYS setup only makes the adjustments listed above to the local machine. If your OPC I/O server runs on a remote machine, you may need to take similar steps on that machine as well.

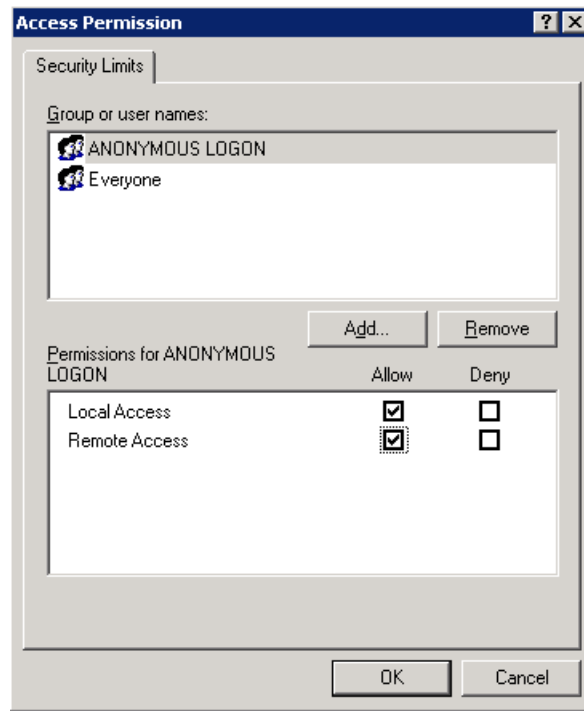
## COM Security

An important step to completing the configuration of DCOM is to set system-wide COM access, launch, and activation permissions. The TrakSYS setup cannot do this automatically because these settings depend on your environment and improper settings may expose your system to security risks.

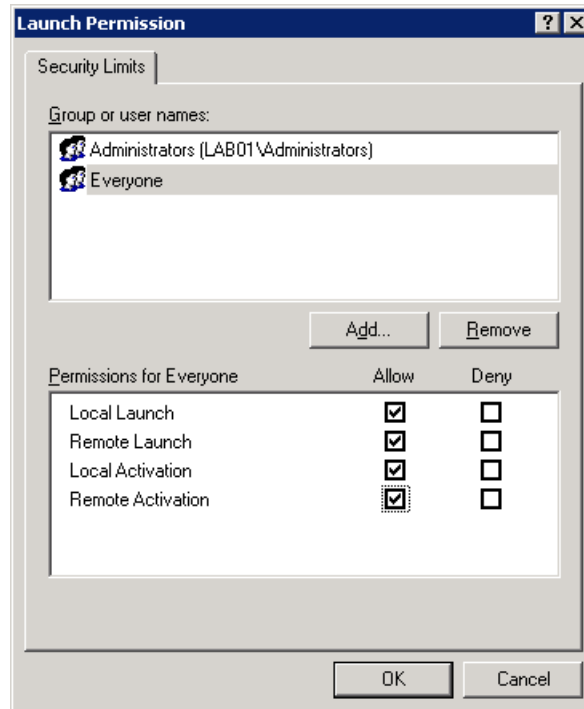
In Windows XP and Windows Server 2003, these settings are located in the Component Services MMC snap-in, located in Administrative Tools.



In the Access Permissions section, click Edit Limits and allow the ANONYMOUS LOGON Remote Access as shown below.



Click OK and then click the Edit Limits button in the Launch and Activation Permissions sections. In this section you should ideally grant Remote Launch and Remote Activation permissions to a special Windows User or Group designated for OPC use, but a simpler approach is to grant these permissions to Everyone as shown below.



Granting these Launch permissions to Everyone should be a temporary measure while troubleshooting OPC. Once OPC communication is working, you should reassign these permissions to a special Windows User or Group dedicated to running OPC software; this precaution will improve security in DCOM. Remember to configure the LOGICTrak Service to use a matching login or OPC communication may fail.

## Tag and Access Name Configuration

If incorrect security settings are not otherwise inhibiting OPC communication, you may simply need to adjust your Tag and Access Name configuration in MODELTrak.

Tags will appear red in the LOGICTrak Client and have bad quality (typically 0) when there is an OPC communication problem. If all Tags belonging to a particular OPC Access Name are bad, then the Access Name itself may be configured incorrectly.

You can verify an Access Name's settings in MODELTrak by clicking the Test Connection button in the Access Name's Properties dialog. If the test fails, recheck the OPC settings and the OPC I/O server application itself and try again. If the test succeeds but OPC still does not work from LOGICTrak, then you are most likely experiencing a DCOM or firewall issue. If you are still unable to connect to a remote OPC server, recheck the settings listed in the TrakSYS Setup OPC Preparation and COM Security sections for possible DCOM problems.

Note that establishing a test connection from MODELTrak does not guarantee that an Access Name will work in LOGICTrak, particularly if the LOGICTrak Service runs on another computer. The service's credentials may lack the necessary privileges for DCOM or the machine hosting it may not be configured for OPC. Refer to the TrakSYS Setup OPC Preparation and COM Security sections to ensure you have configured the LOGICTrak's host correctly.

Once you have verified that your OPC Access Name's settings are correct, continued OPC problems may simply indicate that your OPC Tags are incorrect. If possible, use a third-party OPC client to test one or more of the Item Names. You should also verify that the Access Path property on the OPC Access Name is correct. At runtime, LOGICTrak uses the Access Path as a prefix to the Item Name of every Tag belonging to the Access Name. If the Access Path is incorrect, it will corrupt the Item Name of every Tag using that Access Name.

## ***Index***

### **A**

Access Names.....	27
Database Access Name Queries .....	30
Activation.....	293
Aggregate Tags.....	46
ALERTTrak .....	282
Archive Database .....	288
Areas.....	48
AUDITTrak .....	284
Availability (OEE) .....	297

### **B**

Batch Quality Loss Categories.....	75
Batch Sub-System Parameter Definitions .....	58
Batch Sub-Systems .....	57
Batch Systems.....	51
BATCHTrak.....	287
Boolean Tags.....	31

### **C**

Calculation Tags.....	32
Calculation Units (KPI) .....	304
Check Referential Integrity.....	288
Compare Tags .....	34
Compile all Scripts.....	21
Conditional Event Subscriptions .....	91
Configuration Entities	
Journal Categories .....	74
Configuration entities	
Product Maps	
Product Map Items.....	76
Configuration Entities .....	25
Access Names .....	27
Database Access Name Queries.....	30
Areas .....	48
Batch Quality Loss Categories.....	75
Compile all Scripts .....	21
Custom Properties.....	97
Custom Property Schemes .....	97
Devices .....	89
Event Categories .....	66
Event Category Groups.....	67
Event Category Schemes .....	67
Event Category Scheme Levels.....	67
Event Codes.....	67
Event Definitions .....	59
Event Definition EVENTTrak Node Definitions .....	61
Event Definition Tag Relationships.....	62
EVENTTrak Nodes .....	25

Function Definitions.....	64
Function Definition Parameter Definitions .....	65
Function Definition Tag Relationships .....	65
Job Capture Schemes	
Job Capture Scheme Columns .....	53
Job Capture Tags.....	53
KPI Calculation Views .....	74
KPI Calculations.....	68
KPI Capture Schemes.....	73
KPI Counters.....	71
KPI VTR Definitions.....	72
KPI Count Categories .....	74
LOGICTrak Instances.....	26
Materials.....	81
Material Alternates.....	81
Material Components .....	81
Product Attributes .....	75
Product Maps .....	76
Product Schemes .....	75
Product Sets.....	76
Products .....	75
Recipe Step Definitions	
Recipe Step Definition Parameters .....	79
Recipes.....	76
Recipe Parameters.....	80
Recipe Parameter Ranges.....	80
Recipe Step Definitions	
Recipe Step Definition Materials .....	80
Sample Categories .....	74
Schedule Patterns .....	82
Schedule Pattern Items .....	83
Schedules .....	82
SCRIBETrak Item Group	
KPI Interval List Items .....	87
New Event Items.....	88
SCRIBETrak Item Groups.....	84
Event List Items .....	87
SCRIBETrak Items .....	84
Manual Data Entry Items.....	84
SCRIBETrak Nodes.....	27
SCRIBETrak Item Group Assignments .....	27
Scripts.....	96
Shifts .....	83
Step Categories .....	74
Step Category Groups .....	74
Step Category Schemes.....	74
Step Category Scheme Levels .....	75
Step Codes .....	75
Subscribers .....	88
Subscriptions .....	89
Sub-Systems.....	56
Batch Sub-System Parameter Definitions.....	58
Batch Sub-Systems .....	57

Sub-System EVENTTrak Node Definitions .....	58
System Tag Relationships .....	55
System Capture Schemes .....	
System Capture Scheme Columns .....	54
System Capture Tags .....	54
System Views .....	55
Systems .....	49
Batch Systems .....	51
Discrete Systems .....	50
Job Capture Schemes .....	53
Sample Definitions .....	55
System Capture Schemes .....	53
System EVENTTrak Node Definitions .....	52
Tag Views .....	47
Tags .....	30
Aggregate Tags .....	46
Boolean Tags .....	31
Calculation Tags .....	32
Compare Tags .....	34
Counter Tags .....	35
Database Tags .....	36
Latch Tags .....	35
OPC HDA Tags .....	39
OPC Tags .....	37
Script Tags .....	46
State Tags .....	40
Attribute Reference .....	40
Switch Tags .....	45
Tag History Definitions .....	47
Template Tags .....	47
Virtual Tags .....	46
Task Definitions .....	63
Task Form Items .....	64
Task Form Item Groups .....	64
Teams .....	83
TrakSYS Settings .....	21
Users .....	94
Counter Tags .....	35
Counter Units (KPI) .....	304
Create Audit Triggers .....	291
Create Notification Instance .....	291
Custom Properties .....	97
Custom Property .....	
Custom Property Enums .....	98
Custom Property Enums .....	98
Custom Property Schemes .....	97

## **D**

Database Access Name Queries .....	30
Database Manager .....	287
Archive Database .....	288
Check Referential Integrity .....	288
Create Audit Triggers .....	291

Create Database .....	288
Create Notification Instance .....	291
Delete Audit Triggers.....	291
Delete Data.....	290
Delete Notification Instance .....	291
New Database .....	288
Register Remote Notification Instance .....	292
Repair Logins .....	288
Resync Notification Database.....	291
Update Data Source .....	290
Upgrade Database .....	288
Upgrade Report Files .....	291
Database Security .....	16
Logins .....	16
Users .....	16
Database Tags .....	36
Date and Time Format Reference .....	207
Date and Time Picker.....	206
DbConfig.....	114
Delete Audit Triggers .....	291
Delete Data .....	290
Delete Notification Instance.....	291
Devices.....	89
Discrete Sub-Systems.....	57
Discrete Systems .....	50
Display Units (KPI) .....	304
Distributed Architecture .....	9

## **E**

EDB Setup.....	18
Efficiency Metrics .....	296
Event Categories.....	66
Event Category Groups .....	67
Event Category Scheme Levels .....	67
Event Category Schemes.....	67
Event Category Scheme Levels.....	67
Event Codes .....	67
Event Definition EVENTTrak Node Definitions .....	61
Event Definition Import.....	295
Format .....	295
Wizard.....	295
Event Definition Tag Relationships.....	62
Event Definition VTRs .....	62
Event Definitions.....	59
Event Definition EVENTTrak Node Definitions.....	61
Event Definition Tag Relationships .....	62
Event End Subscriptions.....	91
Event List Items.....	87
Event Start Subscriptions .....	90
EVENTTrak .....	112
Common Properties.....	112
Event Category Picker Control .....	116
Event Data Control.....	125



Event List Control .....	135
Localization .....	190
Oee Data Control .....	143
Product Picker Control .....	156
Sample Applications .....	191
Sample C# EVENTTrak Application .....	191
Sample Sub-Group Data Control .....	183
Tag Data Control .....	165
Tag List Control .....	169
Time Span Format .....	189
Touchboard Control.....	183
EVENTTrak Nodes (Configuration Entities) .....	25

## **F**

Flow .....	302
Flow Off Delay .....	302
Flow On Count .....	302
Flow On Period.....	302
Function Definition Parameter Definitions .....	65
Function Definition Tag Relationships.....	65
Function Definitions .....	64
Function Definition Parameter Definitions.....	65
Function Definition Tag Relationships .....	65

## **G**

Getting Started .....	8
Global Variables .....	266

## **H**

HISTORITrak .....	285
-------------------	-----

## **I**

IMPROVETrak .....	285
Input Units (KPI).....	304
Installation .....	16

## **J**

Job Capture Scheme Columns.....	53
Job Capture Schemes .....	53
Job Capture Scheme Columns .....	53
Job Capture Tags .....	53
Job Capture Tags .....	53
Journal Categories.....	74

## **K**

KPI Calculation Views .....	74
KPI Calculations .....	68
KPI Capture Schemes .....	73
KPI Counters .....	71
KPI VTR Definitions .....	72
KPI Capture Scheme Columns.....	73
KPI Capture Schemes .....	73
KPI Capture Tags .....	73

KPI Count Categories.....	74
KPI Counter Flow .....	302
KPI Counters .....	71
KPI Interval List Items .....	87
KPI Subscriptions .....	92
KPI Unit Conversions .....	302
KPI Units.....	302

## **L**

Latch Tags .....	35
License Manager .....	292
Activation .....	293
Local Variables.....	267
LOGICTrak .....	109
LOGICTrak Client .....	110
LOGICTrak Instances (Configuration Entities) .....	26
LOGICTrak Service .....	109
Logins.....	16

## **M**

Manual Data Entry Items.....	84
Material Alternates .....	81
Material Components .....	81
Materials .....	81
Material Alternates .....	81
Material Components.....	81
Maximum Increment per Scan .....	301
MODELTrak .....	19
Logic Tree .....	25
Tag Finder .....	25

## **N**

New Event Items.....	88
----------------------	----

## **O**

OEE.....	296
Availability .....	297
Calculating.....	296
Implementation .....	297
Performance.....	297
Production Time.....	297
Quality .....	297
Scheduled Time .....	296
OEE KPI Definitions .....	72
OLE DB Tags .....	36
OPC HDA Tags .....	39
OPC Tags .....	37
OPC Troubleshooting Guide .....	304
COM Security .....	305
Setup OPC Preparation .....	304
Tag and Access Name Configuration.....	307
OPC Write-Back .....	304

## **P**

Performance (OEE) .....	297
Priority .....	299
Sub-Priority .....	300
Product Attributes .....	75
Product Map Items .....	76
Product Maps .....	76
Product Map Items .....	76
Product Schemes .....	75
Product Sets .....	76
Production Time (OEE) .....	297
Products .....	75

## **Q**

Quality (OEE) .....	297
---------------------	-----

## **R**

Recipe Duration Ranges .....	78
Recipe Parameter Ranges .....	80
Recipe Parameters .....	80
Recipe Parameter Ranges .....	80
Recipe Step Definition Duration Ranges .....	79
Recipe Step Definition Materials .....	80
Recipe Step Definition Parameter Ranges .....	79
Recipe Step Definition Parameters .....	79
Recipe Step Definition Parameter Ranges .....	79
Recipe Step Definitions .....	77
Recipe Step Definition Duration Ranges .....	79
Recipe Step Definition Materials .....	80
Recipe Step Definition Parameters .....	79
Recipes .....	76
Recipe Duration Ranges .....	78
Recipe Parameters .....	80
Recipe Parameter Ranges .....	80
Recipe Step Definitions .....	77
Recipe Step Definition Materials .....	80
Register Remote Notification Instance .....	292
Repair Logins .....	288
Report Designer .....	206
Report Types .....	192
Report Variables .....	267
Reports .....	206
Administration	
Audit Dump .....	255
Audit Summary .....	254
Journal Summary .....	256
Administration	
Data Range .....	253
KPI Counter Detail Report .....	251
KPI Dump .....	252
Log Summary Report .....	250
Administration .....	250

Dashboards.....	208
Dashboard.....	208
Editors.....	258
Event List Editor.....	258
Event List with Sub Events.....	259
KPI Interval List Editor.....	261
New Event Editor.....	260
New Journal Entry.....	265
New KPI Interval Adjustment Editor.....	262
New KPI Range Adjustment Editor.....	263
Event Reports.....	209
Area Event Summary (Date) Report.....	210
Area Event Summary (DateTime) Report.....	217
Mean Time Between Failure (Date) Report.....	225
System Event Summary (Date) Report.....	213
System Event Summary (DateTime) Report.....	221
Extensibility Reports.....	257
Link Reporting Services Report.....	258
Link URL Report.....	257
KPI Reports.....	227
Availability Detail Report.....	232
KPI Comparison Detail.....	247
OEE Detail Report.....	229
OEE Summary Report.....	227
Performance Detail Report.....	234
Production Loss Report.....	244
Quality Detail Report.....	237
Takt Detail Report.....	242
TEEP Detail Report.....	239
URL Link Refresh Report.....	257
Resync Notification Database.....	291

## **S**

Sample C# EVENTTrak Application.....	191
Sample Categories.....	74
Sample Definitions.....	55
Schedule Pattern Items.....	83
Schedule Patterns.....	82
Schedule Pattern Items.....	83
Scheduled Time (OEE).....	296
Schedules.....	82
SCRIBETrak.....	282
SCRIBETrak Item Group	
KPI Interval List Items.....	87
New Event Items.....	88
SCRIBETrak Item Group Assignments.....	27
SCRIBETrak Item Groups.....	84
Event List Items.....	87
SCRIBETrak Items.....	84
Manual Data Entry Items.....	84
SCRIBETrak Nodes	
SCRIBETrak Item Group Assignment.....	27
SCRIBETrak Nodes (Configuration Entities).....	27

Script Class Templates .....	108
Script Tags .....	46
Scripts .....	96
Setup .....	16
Shifts .....	83
SPCTrak .....	286
State Tag Attribute Reference .....	40
State Tags .....	40
Attribute Reference .....	40
Step Categories .....	74
Step Category Groups .....	74
Step Category Scheme Levels .....	75
Step Category Schemes .....	74
Step Category Scheme Levels .....	75
Step Codes .....	75
Sub-Events .....	299
Sub-Priority .....	300
Subscribers .....	88
Subscriptions .....	89
Conditional Event Subscriptions .....	91
Event End Subscriptions .....	91
Event Start Subscriptions .....	90
KPI Subscriptions .....	92
Tag Subscriptions .....	94
Sub-System EVENTTrak Definitions .....	58
Sub-Systems .....	56
Batch Sub-System Parameter Definitions .....	58
Batch Sub-Systems .....	57
Discrete Sub-Systems .....	57
Sub-System EVENTTrak Node Definitions .....	58
System Tag Relationships .....	55
Supported Operating Systems .....	9
Switch Tags .....	45
System Capture Scheme Columns .....	54
System Capture Schemes .....	53
System Capture Scheme Columns .....	54
System Capture Tags .....	54
System Capture Tags .....	54
System EVENTTrak Node Definitions .....	52
System Tag Relationships .....	55
System Templates .....	99
System Views .....	55
System VTR Definitions .....	54
Systems .....	49
Batch Systems .....	51
Discrete Systems .....	50
Job Capture Schemes .....	53
Sample Definitions .....	55
System Capture Schemes .....	53
System EVENTTrak Node Definitions .....	52

## **T**

Tag Finder .....	25
------------------	----

Tag History Definitions.....	47
Tag Import.....	294
Format .....	294
Wizard.....	294
Tag Subscriptions.....	94
Tag Views .....	47
Tags.....	30
Aggregate Tags .....	46
Boolean Tags.....	31
Calculation Tags .....	32
Compare Tags .....	34
Counter Tags.....	35
Database Tags.....	36
Latch Tags .....	35
OLE DB Tags .....	36
OPC HDA Tags.....	39
OPC Tags.....	37
Script Tags.....	46
State Tags .....	40
Attribute Reference.....	40
Switch Tags .....	45
Tag History Definitions .....	47
Template Tags.....	47
Virtual Tags.....	46
Takt Time.....	298
Task Definition VTRs.....	63
Task Definitions .....	63
Task Form Items.....	64
Task Form Item Enums .....	64
Task Form Item Groups .....	64
Task Form Items.....	64
Task Form Item Enums.....	64
Teams .....	83
Technical Support .....	16
TEEP.....	298
Template Tags .....	47
Touchboard Control .....	183
TrakSYS Settings .....	21

## **U**

Update Data Source .....	290
Upgrade Database.....	288
Upgrade Report Files .....	291
Upgrading From a Previous Version .....	19
Users.....	16, 94

## **V**

Virtual Tags.....	46
VTR Definitions (Event Definition) .....	62
VTR Definitions (System) .....	54
VTR Definitions (Task Definition).....	63

## **W**

WEBTrak	
Administration .....	192
Diagnostics .....	203
Server Information .....	204
User Information .....	204
Local Variables .....	267
Localization .....	268
Parameter Types.....	202
Report Designer.....	206
Report Groups .....	204
Permissions .....	205
Report Programming .....	267
Report Types.....	192
Report Variables .....	267
Reporting Fundamentals .....	192
Reports .....	206
Security.....	191
User Identifiers.....	202
Users .....	201
Variables .....	266
Global Variables .....	266
WEBTrak	
Dashboard Reports .....	268
WEBTrak Knowledge Management Portal.....	191