Technical Document

NiagaraAX Tls (Veeder-Root) Driver Guide

December 6, 2006

NiagaraAX Tls (Veeder-Root) Driver Guide

Copyright © 2006 Tridium, Inc.
All rights reserved.
3951 Westerre Pkwy, Suite 350
Richmond
Virginia
23233
U.S.A.

Copyright Notice

Trademark Notices

NiagaraAX Tls (Veeder-Root) Driver Guide

06 December 2006

This documents usage of the Tls (Veeder-Root) driver for the NiagaraAX framework..

Compatibility

Driver Description

The NiagaraAX Tls driver for the Veeder-Root TLS-250 and -350 products (feature code tls) is a package of software that enables a JACE to communication with one or more Veeder-Root TLS 250 and 350 Controllers. The Veeder-Root TLS 250 and 350 controllers are capable of either an Ethernet or RS-232 connection depending on options included in the Veeder-Root controllers.

Platforms Supported

The Tls driver is supported by all NiagaraAX JACE platforms. The **tls** module is currently available for 3.0.99, and soon to be for 3.1.x.

TLS Function Codes Supported

The NiagaraAX Tls driver currently offers support for the following TLS function codes:

TLS 250

Function Code	Function Type
10	In-Tank Inventory Report

TLS 350

Function Code	Function Type
111	Priority Alarm History Report
201	In-Tank Inventory Report
205	In-Tank Status Report
20B	BIR Adjusted Delivery Report
227	Daily Delivery Variance Report
281	Fuel Management Report

The data can be accessed in one of two ways. The driver objects (such as Tls350PriAlmHis, Tls350BirAdjDel, Tls350InTankInv, Tls350InTankSta, etc) have individual output properties for each retrieved value. In addition, due to the desire of some customers to have the data from these reports available in NiagaraAX history data; these driver objects combine the values into a stringOutput (comma delimited format) with a single timestamp.

Two possible solutions for NiagaraAX applications are:

- The Tls driver can push the report data into a standard NiagaraAX history as a comma delimited String value. A database application can then parse the individual data items from the string log.
- Graphics and Logs can be constructed that link to the various driver object outputs to display and log from the most recent retrieved values.

General Function Code Implementation

Most of the Function Codes are implemented as objects. You can configure when the object data is to be collected for each function code.

The following table indicates the data that will be recorded by each function code.

Function Code	Driver Object	Data Items	Data Items
111	Tls350PriAlmHis	-Alarm/Warning Category -Sensor Category -Alarm Type Number -Tank/Sensor Number	-Alarm State -Date/Time Alarm state occurred -StringOutput
201	Tls350InTankInv	-TimeStamp -Tank -Product -Volume -TC Volume	-Ullage -Height -Water -Temperature -StringOutput
205	Tls350InTankSta	-Tank Number -Read Time -Number of Alarms	-Alarm 1 -Alarm n -StringOutput
20B	Tls350BirAdjDel	-Tank Number -Starting Date/Time -Ending Date/Time -Starting Volume -Ending Volume -Adj Delivery Volume -Adj Temperature Compensated -Delivery Volume -Starting Fuel Height -Starting Fuel Temperature 1 -Starting Fuel Temperature 2 -Starting Fuel Temperature 3 -Starting Fuel Temperature 4 -Starting Fuel Temperature 5 -Starting Fuel Temperature 6	-Ending Fuel Height -Ending Fuel Temperature 1 -Ending Fuel Temperature 2 -Ending Fuel Temperature 3 -Ending Fuel Temperature 4 -Ending Fuel Temperature 5 -Ending Fuel Temperature 6 -Total Dispensed During Delivery -Starting Fuel Temperature Average -Ending Fuel Temperature Average -StringOutput
227	Tls350DelVar	Last Delivery Time	Last Delivery
281	Tls350FuelMgmt	Tank Number Product Code Days Supply of Fuel Remaining Present Inventory Present 95% Ullage Average Sales on Sundays	Average Sales on Mondays Average Sales on Tuesdays Average Sales on Wednesdays Average Sales on Thursdays Average Sales on Fridays Average Sales on Saturdays

Installation

To use the NiagaraAX Tls driver, you must have a target host that is licensed for the “tls” feature. In addition, other device limits or proxy point limits may exist in your license.

From your PC, use the Niagara Workbench 3.*n.nn* installed with the “installation tool” option (checkbox “This instance of Workbench will be used as an installation tool”). This option installs the needed distribution files (.*dist* files) for commissioning various models of remote JACE platforms. The dist files are located under your Niagara install folder in various revision-named subfolders under the “sw” folder.

On your Workbench PC, you also require the following modules:

- **tls**.jar
- **csmgrbase**.jar

Apart from installing the 3.*n.nn* version of the Niagara distribution files in the JACE, make sure to install both of the modules above too (if not already present, or upgrade if an older revision). For more details, see “About the Commissioning Wizard” in the *JACE NiagaraAX Install and Startup Guide*.

Following this, the JACE is now ready for software integration, as described in the rest of this document.

Quick Start

Tls Console

1. If your JACE does not have a Tls Console, add it under its Drivers folder using the **Driver Manager** (double-click Drivers), then choose **New**.

   Select either:
   - **Tls250SerialConsole –** Serial connection to a TLS-250
   - **Tls350 SerialConsole –** Serial connection to a TLS-350
   - **TlsTcpIpConsole –** Ethernet connection to a TLS-350
2. On the property sheet of the added Tls Console, set the following properties:
   - **Station Name Web Supervisor –** Enter the AxSupervisor station name here, if managed by an AxSupervisor (Web Supervisor) application
   - **Comm Type –** Select either Tcpip or Serial depending on your Veeder-Root equipment options.
3. If a TlsTcpIpConsole, set the following properties:
   - **IP address –** Veeder-Root device IP address (e.g. 10.20.1.152).
   - **Port** – Veeder-Root port number (e.g. 10001).
   - **Share Tcpip Connection –** Some users have backup equipment that audits the Veeder-Root devices, and if Comm Type is Tcpip that means those monitoring devices occasionally need to connect to the Tcpip port and query for data. If this is the case, then set this property to “true”—the Tls driver will close its connection after each comm request it does, to make the port accessible to other devices.

4. If a Tls*n*50SerialConsole (Tls250Serial, Tls350Serial), expand the **Serial Port Config** slot and set the following properties:
   - **Port Name** – JACE serial port used, for example COM1. Then other properties below as required.
   - **Baud Rate**
   - **Data Bits**
   - **Stop Bits**
   - **Parity**
   - **Flow Control Mode**
5. Save the station.
6. Reboot after performing the setup above.

Tls Console Device (TLS-350 only)

If connecting to a TLS-350, you can retrieve centralized alarms by adding a "ConsoleDevice" under the Tls350SerialConsole or TlsTcpIpConsole. You can do this from the **Device Manager** (double-click the Tls Console), then choose **New**.

Select: **Tls350ConsoleDevice**

Only one console device should be added for the Veeder-Root system.

Tls Fuel Tank Devices

1. Under the Tls Console, you should add a Tls250FuelTankDevice or Tls350FuelTankDevice component for each fuel tank. You can do this from the **Device Manager** (double-click the Tls Console), then choose **New**.

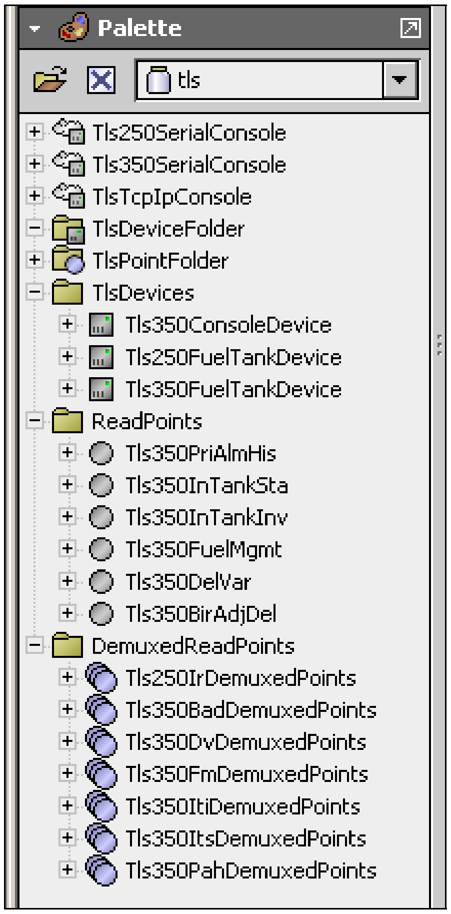
   Select either:
   - **Tls250FuelTankDevice –** If connecting to a TLS-250
   - **Tls350FuelTankDevice –** If connecting to a TLS-350

   Add one FuelTank device per Veeder-Root tank.
2. In the property sheet for each FuelTankDevice, set the following property:
   - **Tank Number –** Enter the Veeder-Root tank number.

Tls350ConsoleDevice and Tls350FuelTankDevice points

Under the Tls350 "device level" ConsoleDevice and FuelTankDevice components, you can copy proxy points under their Points device extension, selecting them from the **tls** *palette* (note that currently, there is no "Points Manager" view on Points containers in the Tls driver).

1. In Workbench, open the tls palette, as shown below.



2. Add the following points from the "ReadPoints" folder in the palette:

Tls350ConsoleDevice Points

Under the Points container of a Tls350ConsoleDevice, add the following:

- **Tls350PriAlmHis** – (Priority Alarm History) Only one per Tls350ConsoleDevice.

Tls350FuelTankDevice Points

Under the Points container of a Tls350FuelTankDevice, add the following:

- **Tls350InTankSta** – One for each fuel tank.
- **Tls350InTankInv** – One for each fuel tank.
- **Tls350BirAdjDel** – One for each fuel tank.

3. In the property sheet of the **Proxy Ext** for each of the points added in step 2, check the following properties and actions:
   - **Minimum Poll Interval**
     - Alarms and status should be 30 sec, 1 min, 60 min, or whatever meets your needs (0 results in no poll).
     - Inventory and delivery should be 15 min, 30 min, 60 min, or whatever meets your need (0 results in no poll).
   - **Actions**
     - **Read Data –** Performs an immediate poll.
     - **Clear Data –** Clears retained data that prevents recording polled events multiple times (use when data also cleared from Veeder-Root controller, so that JACE resources are occasionally freed up).
     - **Clear Logs –** Clears any linked NiagaraAX histories.

Component Guides

Component reference information may be added in a later revision of this document. For now, only a listing of Tls driver components is included.

Networks

Network-level components in the Tls driver are the Tls Console types, namely:

- **Tls250SerialConsole**
- **Tls350SerialConsole**
- **TlsTcpIpConsole**

Devices

Device-level components in the Tls driver are a “DeviceConsole” (TLS-350 only) and “FuelTankDevices”, namely:

- **Tls350ConsoleDevice**
- **Tls250FuelTankDevice**
- **Tls350FuelTankDevice**

Point

Proxy points in the Tls driver include the following types:

- **Tls350BirAdjDel**
- **Tls350DelVar**
- **Tls350FuelMgmt**
- **Tls350InTankInv**

- **Tls350InTankSta**
- **Tls350PriAlmHis**

Demuxed Point Device Extensions

The following "demuxed" point device extensions each contain a frozen PointsFolder with a predefined group of read-only control points, each representing one value:

- **Tls250IrDemuxedPoints**
- **Tls350BadDemuxedPoints**
- **Tls350DvDemuxedPoints**
- **Tls350FmDemuxedPoints**
- **Tls350ItDemuxedPoints**
- **Tls350ItsDemuxedPoints**
- **Tls350PahDemuxedPoints**

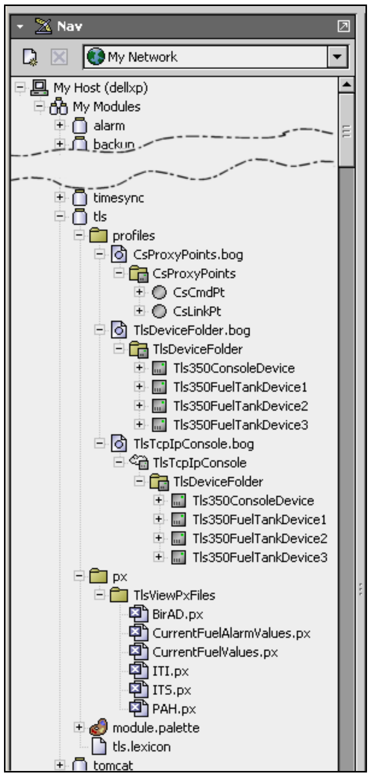
Views

Each of the network-level TlsConsole components has a simple **Device Manager** view, where you can manually add **New** device-level components (the driver has no online "learn" features).

Tls device-level components have Points device extensions, but currently there are *no* Point Manager views. The property sheet and wire sheet views are typically used.

Special Tls Module Features

In addition to components in the tls palette, the tls module contains several configured components that you can access in Workbench by accessing the module using the "My Modules" feature, as shown below.



```
My Host (dellxp)
  My Modules
    alarm
    backup
    timesync
    tls
      profiles
        CsProxyPoints.bog
          CsProxyPoints
            CsCmdPt
            CsLinkPt
        TlsDeviceFolder.bog
          TlsDeviceFolder
            Tls350ConsoleDevice
            Tls350FuelTankDevice1
            Tls350FuelTankDevice2
            Tls350FuelTankDevice3
        TlsTcpIpConsole.bog
          TlsTcpIpConsole
            TlsDeviceFolder
              Tls350ConsoleDevice
              Tls350FuelTankDevice1
              Tls350FuelTankDevice2
              Tls350FuelTankDevice3
      px
        TlsViewPxFiles
          BirAD.px
          CurrentFuelAlarmValues.px
          CurrentFuelValues.px
          ITI.px
          ITS.px
          PAH.px
      module.palette
      tls.lexicon
    tomcat
```

The following main sections describe these tls module items:

- profiles
- px/TlsViewPxFiles

Also see the ending section "History \ station name."

profiles

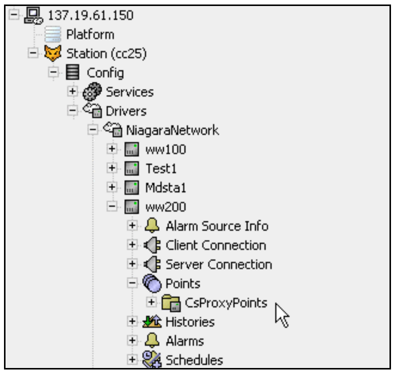
Profiles include the following types:

- CsProxyPoints.bog
- TlsDeviceFolder.bog
- TlsTcpIpConsole.bog

CsProxyPoints.bog

Use this if interested to receive data regarding new Veeder-Root transactions at an AxSupervisor station. It is a component of data transfer between the remote JACE station and the AxSupervisor station.

To use, copy or paste into the Points folder of the NiagaraStation component that represents the AxSupervisor (in the JACE's NiagaraNetwork), as shown below.

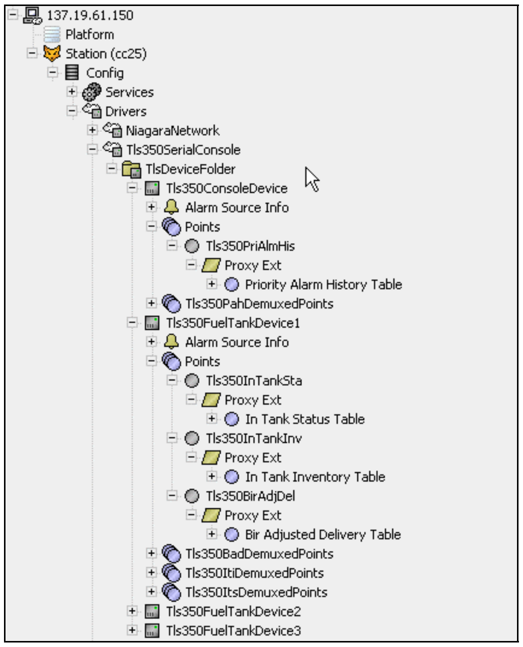


Note that other components are required at the AxSupervisor station, and it may require additional licenses for the **csmgr** application.

TlsDeviceFolder.bog

Use this if interested to configure individual data points for various Veeder-Root components, starting with pre-configured device components and subcomponents (from which you can customize as needed).

To use, copy or paste under the JACE's Drivers\TlsConsole (either of the two valid Tls350Console types, meaning a Tls350SerialConsole or a TlsTcpIpConsole).



As shown in the figure above, the TlsDeviceFolder.bog contains one-preconfigured Tls350ConsoleDevice and 3 Tls350FuelTankDevice components, plus all of their respective child components and pre-configured demuxed points for the following:

- Priority History Alarms
- BIR Adjusted Delivery Transactions

- In-Tank Inventory Transactions
- In-Tank Alarms

You can keep what is relevant and delete what is not, and adjust configurations to match your specific installation.

TlsTcpIpConsole.bog

Use this if interested to configure individual data points for various Veeder-Root components, starting with a pre-configured TlsTcpIpConsole and a TlsDeviceFolder (as described in the previous section "TlsDeviceFolder.bog").

To use, copy or paste under the JACE's Drivers folder. You can keep what is relevant and delete what is not, and adjust configurations to match your specific installation.

px/TlsViewPxFiles

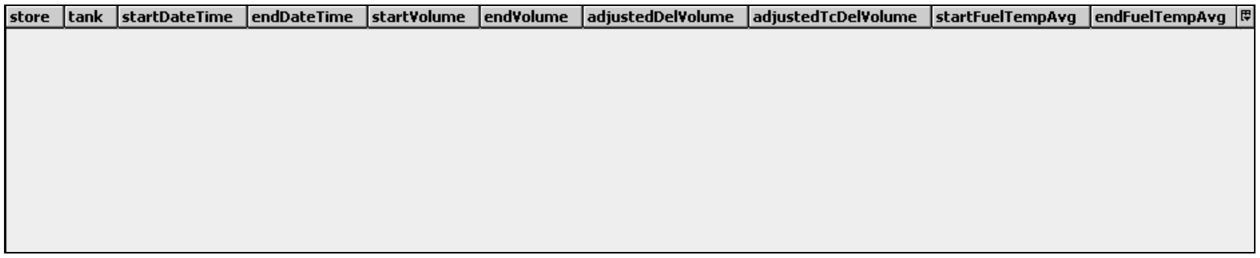
The px folder includes a TlsViewPxFiles subfolder. To use, you copy and paste this subfolder under the JACE station's "Files" folder. The contained Px views are already integrated with components in the TlsDeviceFolder.bog and TlsTcpIpConsole.bog. This lets you right-click on the components, choose Views, and then select the appropriate view from a list.

Included in the TlsViewPxFiles subfolder are the following view types:

- BirAD.px
- CurrentFuelAlarmValues.px
- CurrentFuelValues.px
- ITI.px
- ITS.px
- PAH.px

BirAD.px

This provides a transaction-oriented view of BIR adjusted delivery entries, using 1 row-per-transaction. A button to hyperlink to this display is available on the CurrentFuelValues view. This is also a view on Bir Adjusted Delivery Table.

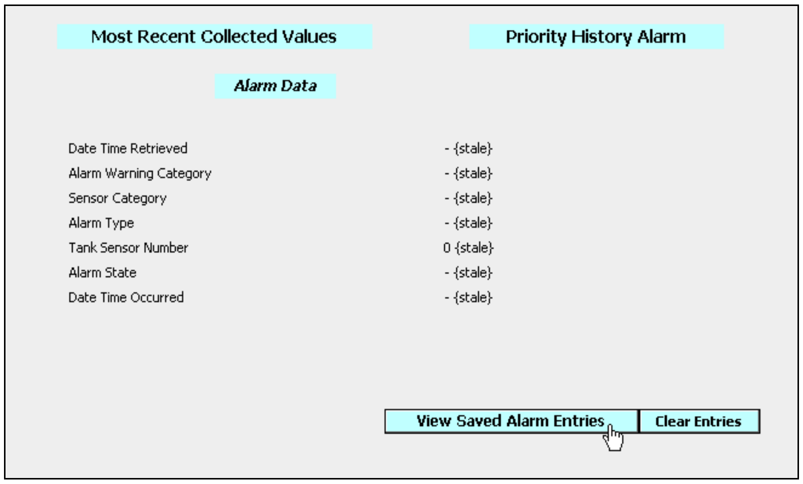


store	tank	startDateTime	endDateTime	startVolume	endVolume	adjustedDelVolume	adjustedTcDelVolume	startFuelTempAvg	endFuelTempAvg

The number of rows available in this view is configurable on the Tls350BirAdjDel point, using its proxyExt property "Max Old Entries Saved" (default is 16).

CurrentFuelAlarmValues.px

This provides a "snapshot" view of the most recent alarm values, plus a button to hyperlink to a BQL query for recent alarm transactions, and another button to clear that BQL query.

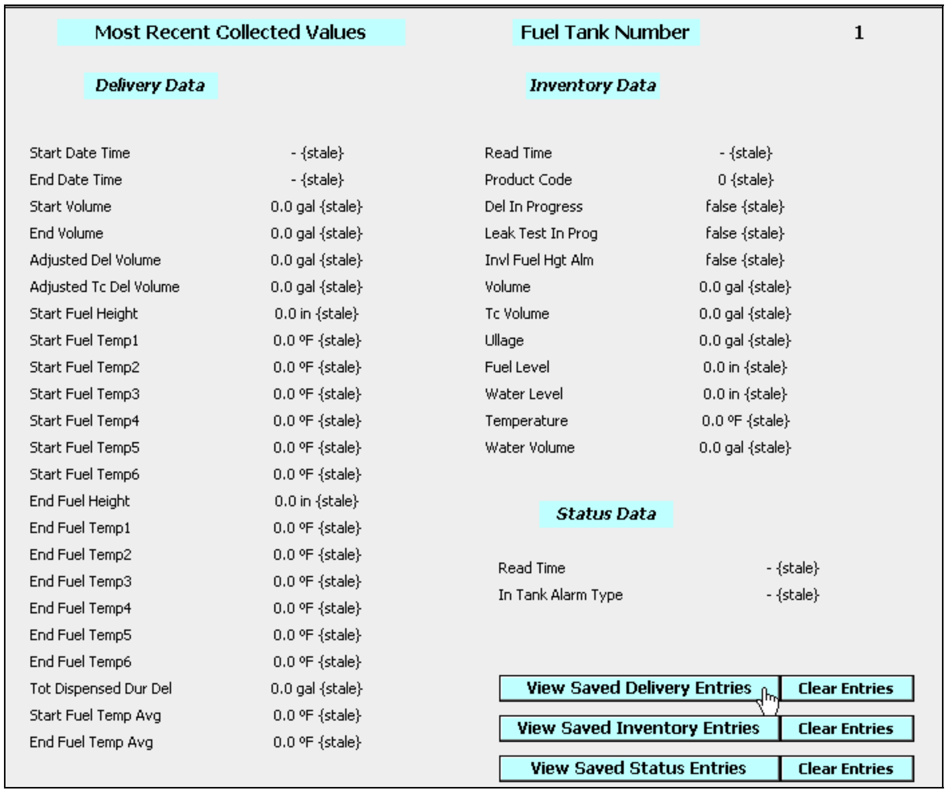


Date Time Retrieved	- {stale}
Alarm Warning Category	- {stale}
Sensor Category	- {stale}
Alarm Type	- {stale}
Tank Sensor Number	0 {stale}
Alarm State	- {stale}
Date Time Occurred	- {stale}

CurrentFuelValues.px

This provides a "snapshot" view of the most recent values, plus buttons to hyperlink to BQL queries of recent value transactions (and also to *clear* those BQL queries) for the following items:

- BIR Adjusted Delivery
- In-Tank Delivery
- In-Tank Status transactions


Most Recent Collected Values
Fuel Tank Number
1
Delivery Data
Inventory Data
Status Data
View Saved Delivery Entries
Clear Entries
View Saved Inventory Entries
Clear Entries
View Saved Status Entries
Clear Entries

Delivery Data

Start Date Time	- {stale}
End Date Time	- {stale}
Start Volume	0.0 gal {stale}
End Volume	0.0 gal {stale}
Adjusted Del Volume	0.0 gal {stale}
Adjusted Tc Del Volume	0.0 gal {stale}
Start Fuel Height	0.0 in {stale}
Start Fuel Temp1	0.0 ºF {stale}
Start Fuel Temp2	0.0 ºF {stale}
Start Fuel Temp3	0.0 ºF {stale}
Start Fuel Temp4	0.0 ºF {stale}
Start Fuel Temp5	0.0 ºF {stale}
Start Fuel Temp6	0.0 ºF {stale}
End Fuel Height	0.0 in {stale}
End Fuel Temp1	0.0 ºF {stale}
End Fuel Temp2	0.0 ºF {stale}
End Fuel Temp3	0.0 ºF {stale}
End Fuel Temp4	0.0 ºF {stale}
End Fuel Temp5	0.0 ºF {stale}
End Fuel Temp6	0.0 ºF {stale}
Tot Dispensed Dur Del	0.0 gal {stale}
Start Fuel Temp Avg	0.0 ºF {stale}
End Fuel Temp Avg	0.0 ºF {stale}

Inventory Data

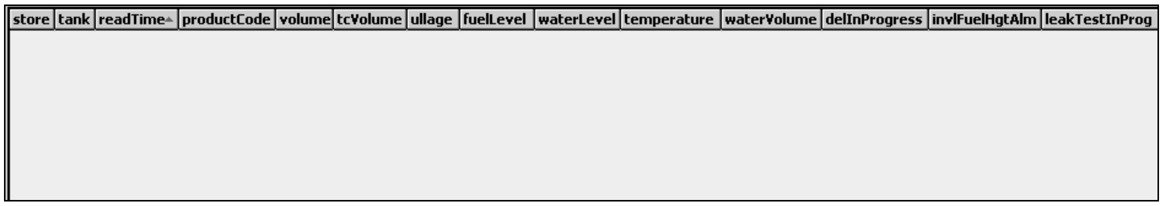
Read Time	- {stale}
Product Code	0 {stale}
Del In Progress	false {stale}
Leak Test In Prog	false {stale}
Invl Fuel Hgt Alm	false {stale}
Volume	0.0 gal {stale}
Tc Volume	0.0 gal {stale}
Ullage	0.0 gal {stale}
Fuel Level	0.0 in {stale}
Water Level	0.0 in {stale}
Temperature	0.0 ºF {stale}
Water Volume	0.0 gal {stale}

Status Data

Read Time	- {stale}
In Tank Alarm Type	- {stale}

ITI.px

This provides a transaction-oriented view of In-Tank Inventory entries, using 1 row-per-transaction. A button to hyperlink to this display is available on the CurrentFuelValues view. This is also a view on the In-Tank Inventory Table.

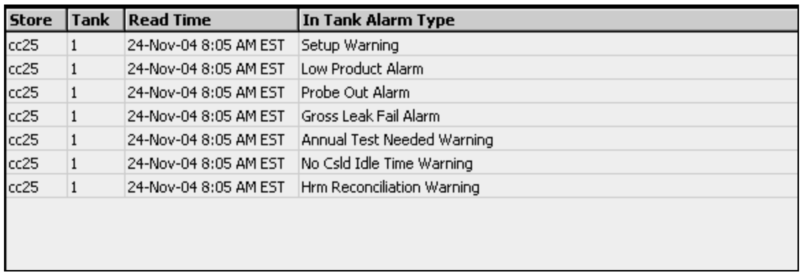


store	tank	readTime	productCode	volume	tcVolume	ullage	fuelLevel	waterLevel	temperature	waterVolume	delInProgress	invlFuelHgtAlm	leakTestInProg

The number of rows available in this view is configurable on the Tls350InTankInv point, using its proxyExt property “Max Old Entries Saved” (default is 16).

ITS.px

This provides a transaction-oriented view of In-Tank Status entries, using 1 row-per-transaction. A button to hyperlink to this display is available on the CurrentFuelValues view. This is also a view on the In-Tank Status Table.

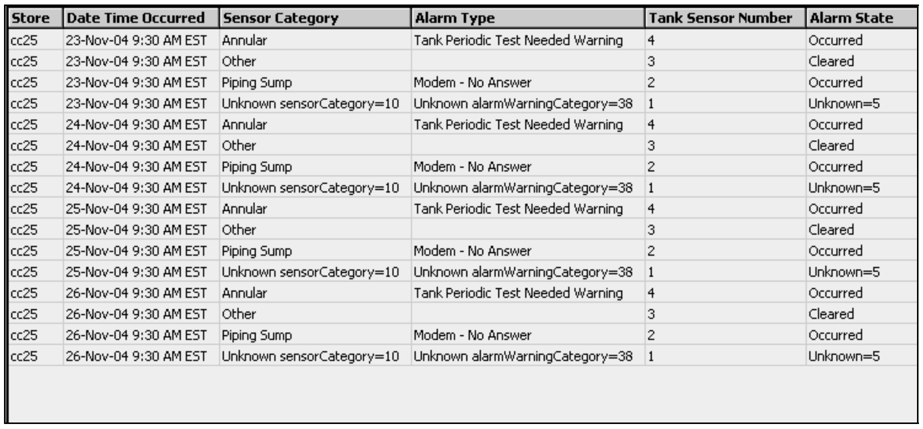


Store	Tank	Read Time	In Tank Alarm Type
cc25	1	24-Nov-04 8:05 AM EST	Setup Warning
cc25	1	24-Nov-04 8:05 AM EST	Low Product Alarm
cc25	1	24-Nov-04 8:05 AM EST	Probe Out Alarm
cc25	1	24-Nov-04 8:05 AM EST	Gross Leak Fail Alarm
cc25	1	24-Nov-04 8:05 AM EST	Annual Test Needed Warning
cc25	1	24-Nov-04 8:05 AM EST	No Csld Idle Time Warning
cc25	1	24-Nov-04 8:05 AM EST	Hrm Reconciliation Warning

The number of rows available in this view is configurable on the Tls350InTankSta point, using its proxyExt property “Max Old Entries Saved” (default is 16).

PAH.px

This provides a transaction-oriented view of Priority History Alarm entries, using 1 row-per-transaction. A button to hyperlink to this display is available on the CurrentFuelAlarmValues view. This is also a view on the Priority History Alarm Table.



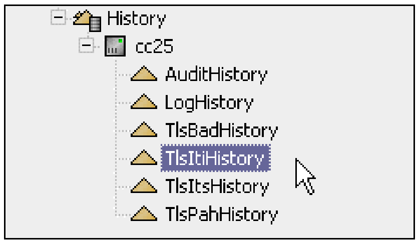
Store	Date Time Occurred	Sensor Category	Alarm Type	Tank Sensor Number	Alarm State
cc25	23-Nov-04 9:30 AM EST	Annular	Tank Periodic Test Needed Warning	4	Occurred
cc25	23-Nov-04 9:30 AM EST	Other		3	Cleared
cc25	23-Nov-04 9:30 AM EST	Piping Sump	Modem - No Answer	2	Occurred
cc25	23-Nov-04 9:30 AM EST	Unknown sensorCategory=10	Unknown alarmWarningCategory=38	1	Unknown=5
cc25	24-Nov-04 9:30 AM EST	Annular	Tank Periodic Test Needed Warning	4	Occurred
cc25	24-Nov-04 9:30 AM EST	Other		3	Cleared
cc25	24-Nov-04 9:30 AM EST	Piping Sump	Modem - No Answer	2	Occurred
cc25	24-Nov-04 9:30 AM EST	Unknown sensorCategory=10	Unknown alarmWarningCategory=38	1	Unknown=5
cc25	25-Nov-04 9:30 AM EST	Annular	Tank Periodic Test Needed Warning	4	Occurred
cc25	25-Nov-04 9:30 AM EST	Other		3	Cleared
cc25	25-Nov-04 9:30 AM EST	Piping Sump	Modem - No Answer	2	Occurred
cc25	25-Nov-04 9:30 AM EST	Unknown sensorCategory=10	Unknown alarmWarningCategory=38	1	Unknown=5
cc25	26-Nov-04 9:30 AM EST	Annular	Tank Periodic Test Needed Warning	4	Occurred
cc25	26-Nov-04 9:30 AM EST	Other		3	Cleared
cc25	26-Nov-04 9:30 AM EST	Piping Sump	Modem - No Answer	2	Occurred
cc25	26-Nov-04 9:30 AM EST	Unknown sensorCategory=10	Unknown alarmWarningCategory=38	1	Unknown=5

The number of rows available in this view is configurable on the Tls350PriAlmHis point, using the proxyExt property "Max Old Entries Saved" (default is 16).

History \ station name

For each polled point type (Tls350BirAdjDel, Tls350InTankInv, Tls350InTankSta, Tls350PriAlmHis, Tls350FuelMgmt, and Tls350DelVar), histories are automatically created, and a new history record is saved upon each new poll transaction received.

These histories are preconfigured with capacity of 500 records and a full policy of "roll." You can view them by expanding the station's History folder and then double-clicking on the history of interest (for example, TlsPahHistory).



Maintenance can be run from the Database Maintenance view on the History component, and backups can be configured using standard NiagaraAX tools.

The demuxed point type values (all with a Tls350DemuxedProxyExt) are derived from the polled point types, and thus are not recorded in history records discretely. These values only have meaning in the context of a transaction, such as the most recent transaction, or the most recent transaction minus 1, and so on.