

SolventTrak Remote Interface with Firmware Version 1.09 or later

SolventTrak provides two separate ways to “validate” peak detection activity. First is the analog Signal Out which provides tick marks to indicate peak detection activity. Second is an RS232 interface which can provide real time peak detection information which can be logged into a LIMS system or other program in order to establish a log of daily activity. The RS232 interface will be described below.

In addition to logging, up to 15 method parameter files may be downloaded to SolventTrak from a PC. Axxiom Chromatography provides an example method file editor program to create and download a method. Customers may create their own programs by using the information provided in this document.

RS232 Serial Interface

Serial communication requires a baud rate of 19,200, no parity, 8 data bits, and 1 stop bits.

Axxiom Chromatography offers an optional communications cable. The cable has a standard DB9 Serial cable on the PC end and an RJ11 plug on the SolventTrak end. The rear panel of the SolventTrak provides a jack to plug the cable into.

Codes and establishing communications

In order to commence communications the PC program or LIMS system must send the character “D” to the SolventTrak. When a “D” is received, the SolventTrak will echo back with an “R” followed by a sign on block, and then it will commence to send data. The form of the data is described next.

D = Send Sign on header, then data codes.

Header = "ST ONLINE, nnnnnnn,
Vvvvv,fff,sssss,wwwwww,dddddd,ttttt,cccc,aa,ttttt,"

where ...

nnnnnnn is the serial number of the unit

vvvv is firmware version number

fff is file number is use

sssss is slope parameter

wwwwww is Width parameter

dddddd is Delay parameter

ttttt is Tick Height parameter

cccc is CleanUpTime parameter

aa is Alarm 0=mute, 1= On

ttttt is Threshold for threshold mode.

Data Codes sent to PC program from SolventTrak

B = Begin Peak

E = End Peak

V = Valve switched to Waste

v = Valve switched to Recycle

. = Tick every one second when no other code is sent

z = external AutoZero command received

T = Threshold exceeded

t = Threshold reset

By logging the constant stream of codes sent by SolventTrak, a validation trail may be established. When no peak start or end is detected an ellipse character will be sent at one second intervals. These tick characters can be used to determine the passage of time and to ascertain that the unit is working. Should a power outage take place the logging program can detect that by the absence of tick characters and then send the "D" character to restart the sending of data.

Method File Format

All method files must contain exactly 41 ASCII characters. When a value is less than the width of that parameter it is to be filled out with space characters. Here is an example of a method file...

```
M15,14400,300,99,1600,999,0,1000000,49350
```

“M” is the Command code to indicate a method file is being sent
“15” is the file number. Range is 1 to 15. This field is 2 characters wide.
“,” is a delimiter to separate one field from the next and is required.
“14400” is the slope parameter. Range is 15 to 14400. Field is 5 characters.
“300” is Width parameter. Range is 1 to 300. Field is 3 characters.
“99” is Delay parameter. Range is 0 to 99. Field is 2 characters.
“1600” is Tick Height parameter. Range is 0 to 6500 and is per user manual list.
“999” is CleanUpTime parameter. Range is 0 to 999. Field is 3 characters wide.
“0” is Alarm parameter. Range is 0 -1. 0 is mute, 1= On. Field is one character
“1000000” is Threshold parameter. Range 0 – 1000000. Field is 7 characters
“49350” is Checksum. CRC-16. Field is 5 characters.
NOTE These fields total up to exactly 41 characters.

When a method file is sent to SolventTrak it will be validated in two ways. First the checksum will be verified and if correct an **ACK** character will be sent. If it fails a **NAK** character will be sent and the file will be ignored. A second check will follow to verify that all parameters are in valid range. If so a second **ACK** character will be sent, otherwise a **NAK** character will be sent. An ASCII **ACK** has the value of **6** decimal. An ASCII **NAK** has the value of **21** decimal.

Setting the Current Method File and Locking Front Panel

In order to set the current file to be used, a “#” character is sent, followed by the file number. This command must contain exactly 3 character. A space character would be used to fill out the number field if less than two digits are required. An example of selecting file 15 follows...

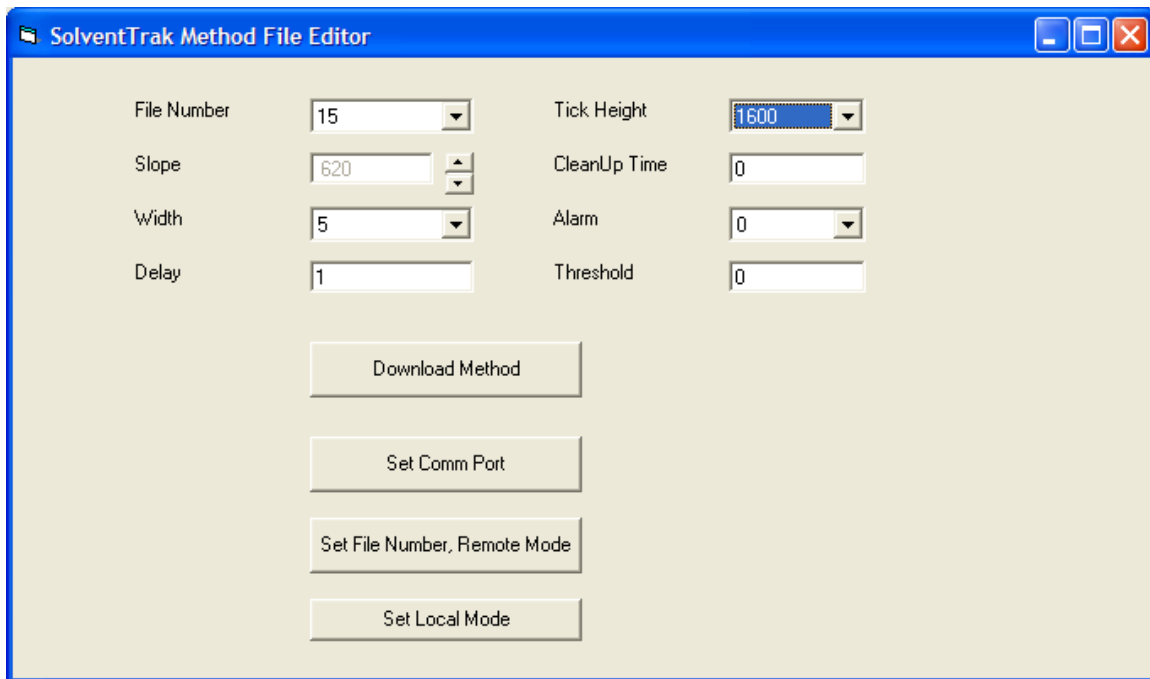
```
#15
```

When a file command is received by SolventTrak it will lock the front panel keypad out and the display will show a two digit file number followed by an “*” in the top left display area. When you observe the file number and “*” character you will know it is in the remote mode and the keypad is locked.

Stored Method files number from 1 to 15. File # 00 is reserved to place the SolventTrak into local front panel mode. Sending #00 will clear the remote mode and unlock the keypad.

Demo PC Method File Editor Program

The Axxiom Chromatography Demo Program is shown below. You may set the parameters for the file, along with file number. Also select the PC port to use for communications and then download the file. Up to 15 files ranging from file number 1-15 may be created and downloaded. Then you may send the command to set the desired file to use.



The screenshot shows a window titled "SolventTrak Method File Editor". The interface contains several input fields and buttons. The fields are arranged in two columns:

Parameter	Value
File Number	15
Slope	620
Width	5
Delay	1
Tick Height	1600
CleanUp Time	0
Alarm	0
Threshold	0

Below the input fields are four buttons:

- Download Method
- Set Comm Port
- Set File Number, Remote Mode
- Set Local Mode

CRC-16 Checksum Calculation in Visual Basic

```
Private aCRC16Table(255) As Integer
```

```
Public Function crc16(sMessage As String) As Integer
```

```
' Given table is already setup.
```

```
' Set iCRC = 0
```

```
' For each byte in message
```

```
' calculate iCRC = (iCRC >> 8) ^ Table[(iCRC & 0xFF) ^ byte]
```

```
' Return iCRC
```

```
Dim iCRC As Integer
```

```
Dim i As Long
```

```
Dim bytT As Byte
```

```
Dim bytC As Byte
```

```
Dim ia As Integer
```

```
iCRC = 0
```

```
For i = 1 To Len(sMessage)
```

```
    bytC = Asc(Mid(sMessage, i, 1))
```

```
    bytT = (iCRC And &HFF) Xor bytC
```

```
    ia = uiShiftRightBy8(iCRC)
```

```
    iCRC = ia Xor aCRC16Table(bytT)
```

```
Next
```

crc16 = iCRC

End Function

```

Public Function uiShiftRightBy8(X As Integer) As Integer
' Shift 16-bit integer value to right by 8 bits
' Avoiding problem with sign bit
Dim iNew As Integer
iNew = (X And &H7FFF) \ 256
If (X And &H8000) <> 0 Then
    iNew = iNew Or &H80
End If
uiShiftRightBy8 = iNew
End Function

Public Function CRC16Setup()

Dim vntA As Variant
Dim i As Integer

' Use variant array to set up table
vntA = Array( _
    &H0, &HC0C1, &HC181, &H140, &HC301, &H3C0, &H280, &HC241, _
    &HC601, &H6C0, &H780, &HC741, &H500, &HC5C1, &HC481, &H440, _
    &HCC01, &HCC0, &HD80, &HCD41, &HF00, &HCFC1, &HCE81, &HE40, _
    &HA00, &HCAC1, &HCB81, &HB40, &HC901, &H9C0, &H880, &HC841, _
    &HD801, &H18C0, &H1980, &HD941, &H1B00, &HDBC1, &HDA81, &H1A40, _
    &H1E00, &HDEC1, &HDF81, &H1F40, &HDD01, &H1DC0, &H1C80, &HDC41, _
    &H1400, &HD4C1, &HD581, &H1540, &HD701, &H17C0, &H1680, &HD641, _
    &HD201, &H12C0, &H1380, &HD341, &H1100, &HD1C1, &HD081, &H1040)

For i = 0 To 63
    aCRC16Table(i) = vntA(i - 0)
Next

vntA = Array( _
    &HF001, &H30C0, &H3180, &HF141, &H3300, &HF3C1, &HF281, &H3240, _
    &H3600, &HF6C1, &HF781, &H3740, &HF501, &H35C0, &H3480, &HF441, _
    &H3C00, &HFCC1, &HFD81, &H3D40, &HFF01, &H3FC0, &H3E80, &HFE41, _
    &HFA01, &H3AC0, &H3B80, &HFB41, &H3900, &HF9C1, &HF881, &H3840, _
    &H2800, &HE8C1, &HE981, &H2940, &HEB01, &H2BC0, &H2A80, &HEA41, _
    &HEE01, &H2EC0, &H2F80, &HEF41, &H2D00, &HEDC1, &HEC81, &H2C40, _
    &HE401, &H24C0, &H2580, &HE541, &H2700, &HE7C1, &HE681, &H2640, _
    &H2200, &HE2C1, &HE381, &H2340, &HE101, &H21C0, &H2080, &HE041)

For i = 64 To 127
    aCRC16Table(i) = vntA(i - 64)
Next

vntA = Array( _
    &HA001, &H60C0, &H6180, &HA141, &H6300, &HA3C1, &HA281, &H6240, _
    &H6600, &HA6C1, &HA781, &H6740, &HA501, &H65C0, &H6480, &HA441, _
    &H6C00, &HACC1, &HAD81, &H6D40, &HAF01, &H6FC0, &H6E80, &HAE41, _
    &HAA01, &H6AC0, &H6B80, &HAB41, &H6900, &HA9C1, &HA881, &H6840, _
    &H7800, &HB8C1, &HB981, &H7940, &HBB01, &H7BC0, &H7A80, &HBA41, _
    &HBE01, &H7EC0, &H7F80, &HBF41, &H7D00, &HBDC1, &HBC81, &H7C40, _
    &HB401, &H74C0, &H7580, &HB541, &H7700, &HB7C1, &HB681, &H7640, _
    &H7200, &HB2C1, &HB381, &H7340, &HB101, &H71C0, &H7080, &HB041)

For i = 128 To 191
    aCRC16Table(i) = vntA(i - 128)
Next

vntA = Array( _
    &H5000, &H90C1, &H9181, &H5140, &H9301, &H53C0, &H5280, &H9241, _
    &H9601, &H56C0, &H5780, &H9741, &H5500, &H95C1, &H9481, &H5440, _
    &H9C01, &H5CC0, &H5D80, &H9D41, &H5F00, &H9FC1, &H9E81, &H5E40, _
    &H5A00, &H9AC1, &H9B81, &H5B40, &H9901, &H59C0, &H5880, &H9841, _
    &H8801, &H48C0, &H4980, &H8941, &H4B00, &H8BC1, &H8A81, &H4A40, _
    &H4E00, &H8EC1, &H8F81, &H4F40, &H8D01, &H4DC0, &H4C80, &H8C41, _
    &H4400, &H84C1, &H8581, &H4540, &H8701, &H47C0, &H4680, &H8641, _
    &H8201, &H42C0, &H4380, &H8341, &H4100, &H81C1, &H8081, &H4040)

```

```
For i = 192 To 255
    aCRC16Table(i) = vntA(i - 192)
Next
```

End Function

Public Function TestCRC16()

```
' Test suite answers:
' CRC16('123456789') = BB3D
' CRC16('hello world')=39C1
' CRC16('Hello world')=F96A
' CRC16('a') = E8C1
' CRC16(' ') = D801
' CRC16('M15,14400,300,99,1600,999,0,1000000,')=49350
```

```
Dim sMessage As String
Dim iCRC As Integer
```

Call CRC16Setup

```
sMessage = "123456789"
iCRC = crc16(sMessage)
Debug.Print "crc16(" & sMessage & ")=" & Hex(iCRC)
```

```
sMessage = "hello world"
iCRC = crc16(sMessage)
Debug.Print "crc16(" & sMessage & ")=" & Hex(iCRC)
```

```
sMessage = "Hello world"
iCRC = crc16(sMessage)
Debug.Print "crc16(" & sMessage & ")=" & Hex(iCRC)
```

```
sMessage = "a"
iCRC = crc16(sMessage)
Debug.Print "crc16(" & sMessage & ")=" & Hex(iCRC)
```

```
sMessage = " " ' (space char)
iCRC = crc16(sMessage)
Debug.Print "crc16(" & sMessage & ")=" & Hex(iCRC)
```

```
sMessage = "M15,14400,300,99,1600,999,0,1000000,"
iCRC = crc16(sMessage)
Debug.Print "crc16(" & sMessage & ")=" & Hex(iCRC) 'shown in hex
Debug.Print "crc16(" & sMessage & ")=" & HexToLong(Hex(iCRC)) 'shown in decimal which is what we want!
```

End Function