TYX CORPORATION

Productivity Enhancement Systems

PAWS Studio Release Notes

Version 1.39.3
April 21, 2011

Table of contents

1 Paws Developer's Studio

Version 1.39.3

Release date: April 21, 2011

1.1 Critical Items

1.2 Known Limitations

1.3 Enhancements

1.3.1 1641CLCEx.exe recognizes the Reset all TPL instruction (ChangeReqID 750)

The following 1641 carrier language code:

```
class ResetAll
{
    public void main()
    {
        <TPL>
        Reset all;
        </TPL>
    }
}
```

gets translated into:

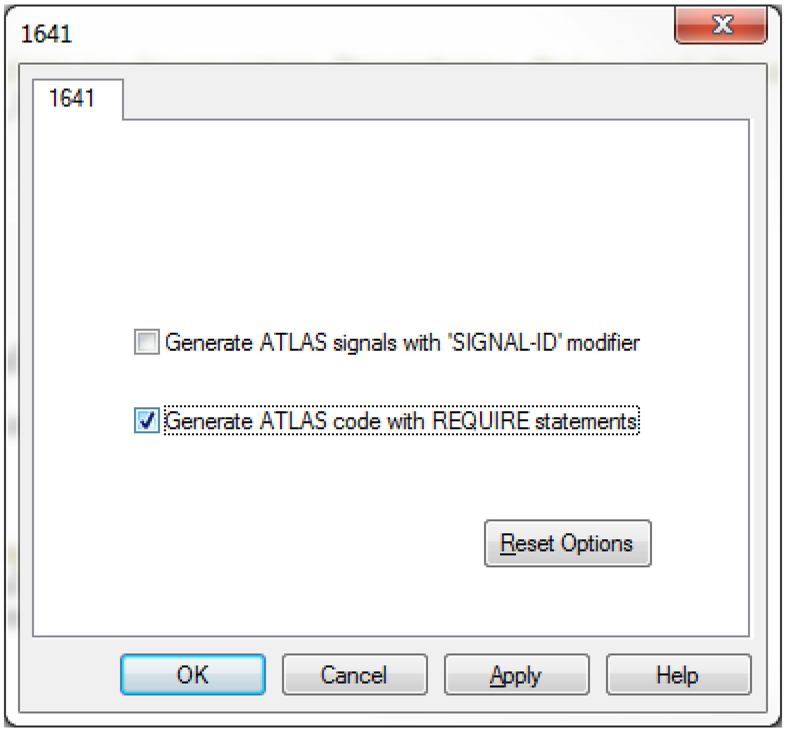
```
        DEFINE, 'ResetAllmain', PROCEDURE
            RESULT
            ('_this_' IS 'ResetAll') $
            REMOVE, ALL $
        END, 'ResetAllmain' $
```

1.3.2 1641CLCEx.exe generates REQUIRE statements in the ATLAS code (TaskID 756)

When the resulting signals in the atlas code are very similar, and they need to be allocated to distinctive instrument whose capabilities are similar or identical, a need to identify the atlas signals from TPL rises.

This is done by the insertion of the REQUIRE statements.

In order to enable the REQUIRE statements, the *Generate ATLAS code with REQUIRE statements* in the *Settings* tab of the 1641 module has to be checked.



- [ ] Generate ATLAS signals with 'SIGNAL-ID' modifier
- [x] Generate ATLAS code with REQUIRE statements

The following 1641 carrier language code

```
class A
{
  public void main()
  {
    <TPL>
    Setup
    <Signal Out="tw1 tw2">
      <TwoWire name='tw1' hi='J1' lo='J2' channelWidth='1' In='c1'/>
      <Constant name='c1' amplitude='5 V' />
      <TwoWire name='tw2' hi='J3' lo='J4' channelWidth='1' In='c2'/>
      <Constant name='c2' amplitude='5 V' />
    </Signal>
    as s1;
    </TPL>
  }
}
```

gets translated into:

```
        REQUIRE, 's1_00', SOURCE, DC SIGNAL,
            CONTROL,
                VOLTAGE 5 V,
            CNX ATLAS-SGN-P-HI ATLAS-SGN-P-LO $
        REQUIRE, 's1_01', SOURCE, DC SIGNAL,
            CONTROL,
                VOLTAGE 5 V,
            CNX ATLAS-SGN-P-HI ATLAS-SGN-P-LO $
```

```
[…]
C The following 's1.c1' 1641 signal is source and its 'Constant' 1641 configuration
type has been mapped to the 'DC SIGNAL' atlas noun. $
            SETUP, DC SIGNAL USING 's1_00',
                VOLTAGE 5 V,
                CNX ATLAS-SGN-P-HI J1 ATLAS-SGN-P-LO J2 $
[…]
C The following 's1.c2' 1641 signal is source and its 'Constant' 1641 configuration
type has been mapped to the 'DC SIGNAL' atlas noun. $
            SETUP, DC SIGNAL USING 's1_01',
                VOLTAGE 5 V,
                CNX ATLAS-SGN-P-HI J3 ATLAS-SGN-P-LO J4 $
```

The name of the atlas signal is composed of the name of the 1641 signal (*s1*), followed by an underscore character and a two decimal digits field that identifies the order of the *configured signal* in the anonymous TSF model.

The name of the atlas signal becomes handy if the developer intends to allocate a specific instrument to it. This is done through an entry in a project included *allocinp* file that associates the name of the atlas signal as described above with a device name and a function number, the latter two identifying the specific instrument.

It is important to understand that the allocinp entries are needed if only the default signal allocation does not satisfy the developer's expectations. Such a circumstance may occur when many identical *configured signals* need to be allocated to distinctive identical instruments.

The name of the atlas signal can also be controlled by an identifier inserted between the *as* keyword and the signal name, in the *Setup* TPL instruction.

The following TPL sequence uses *ps2100* as a signal identifier.

```
class A
{
  public void main()
  {
    <TPL>
    Setup
    <Signal Out="tw1 tw2">
      <TwoWire name='tw1' hi='J1' lo='J2' channelWidth='1' In='c1'/>
      <Constant name='c1' amplitude='5 V' />
      <TwoWire name='tw2' hi='J3' lo='J4' channelWidth='1' In='c2'/>
      <Constant name='c2' amplitude='5 V' />
    </Signal>
    as ps2100 s1;
    </TPL>
  }
}
```

Its atlas translation follows.

```
        REQUIRE, 'ps2100_00', SOURCE, DC SIGNAL,
            CONTROL,
                VOLTAGE 5 V,
            CNX ATLAS-SGN-P-HI ATLAS-SGN-P-LO $
        REQUIRE, 'ps2100_01', SOURCE, DC SIGNAL,
            CONTROL,
                VOLTAGE 5 V,
            CNX ATLAS-SGN-P-HI ATLAS-SGN-P-LO $
[…]
```

```
C The following 's1.c1' 1641 signal is source and its 'Constant' 1641 configuration
type has been mapped to the 'DC SIGNAL' atlas noun. $
            SETUP, DC SIGNAL USING 'ps2100_00',
                VOLTAGE 5 V,
                CNX ATLAS-SGN-P-HI J1 ATLAS-SGN-P-LO J2 $
[…]
C The following 's1.c2' 1641 signal is source and its 'Constant' 1641 configuration
type has been mapped to the 'DC SIGNAL' atlas noun. $
            SETUP, DC SIGNAL USING 'ps2100_01',
                VOLTAGE 5 V,
                CNX ATLAS-SGN-P-HI J3 ATLAS-SGN-P-LO J4 $
```

Unlike the previous case, the atlas signal names are `ps2100_00` and `ps2100_01`, where the explicit identifier `ps2100` replaced the signal name.

This identifier must contain letters and digits only, the first character must be a letter, and its maximum length is 13. The resulting atlas signal name must not exceed 16 characters.

1.4 Problem Reports

1.4.1 PR 11003 Product Licensing Functionality

This feature has been provided starting with this version.

2 Run Time System

Version 1.39.3

Release date: April 21, 2011

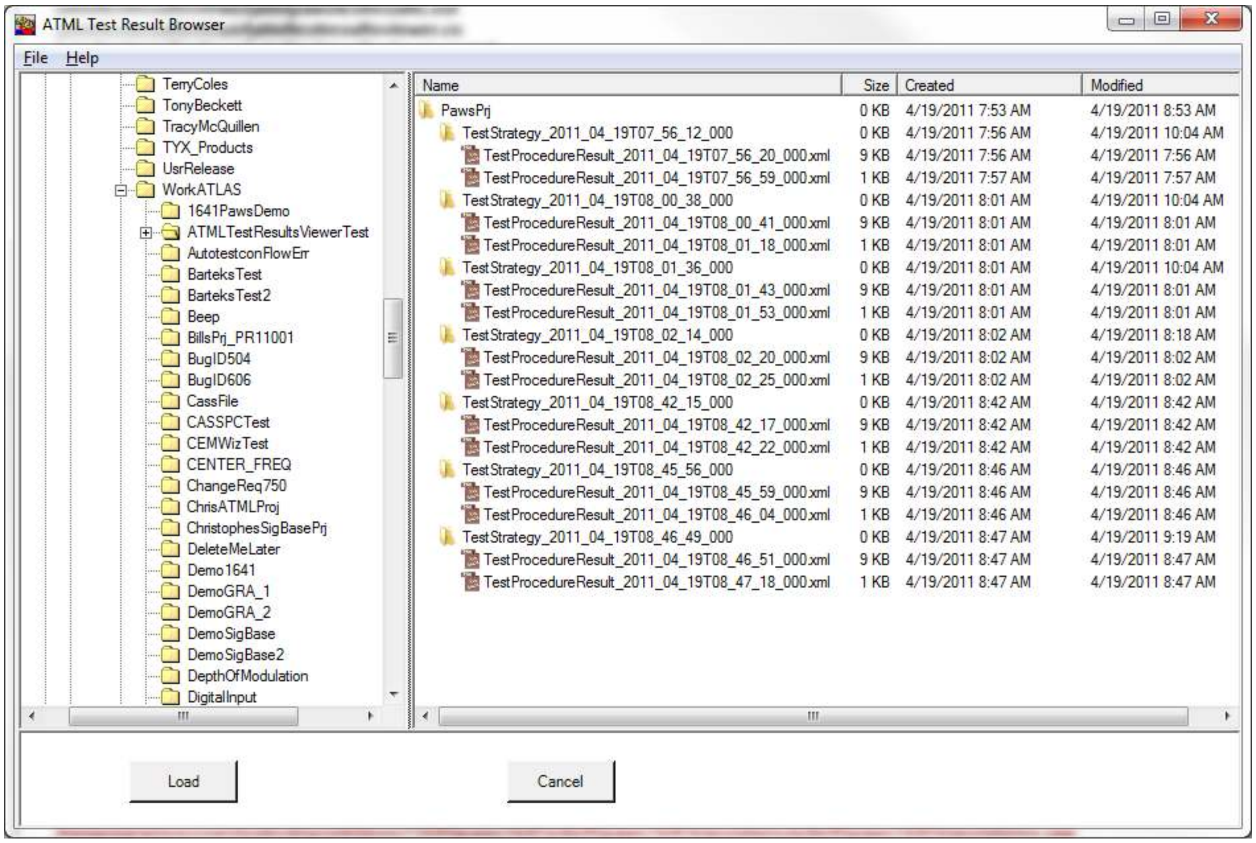
2.1 Critical Items

2.2 Known Limitations

2.3 Enhancements

2.3.1 ATML Test Results Viewer present in <usr>\tyx\datalog

This viewer needs .NET Framework 2.0 to work. It shows the ATML test results grouped under test strategies.



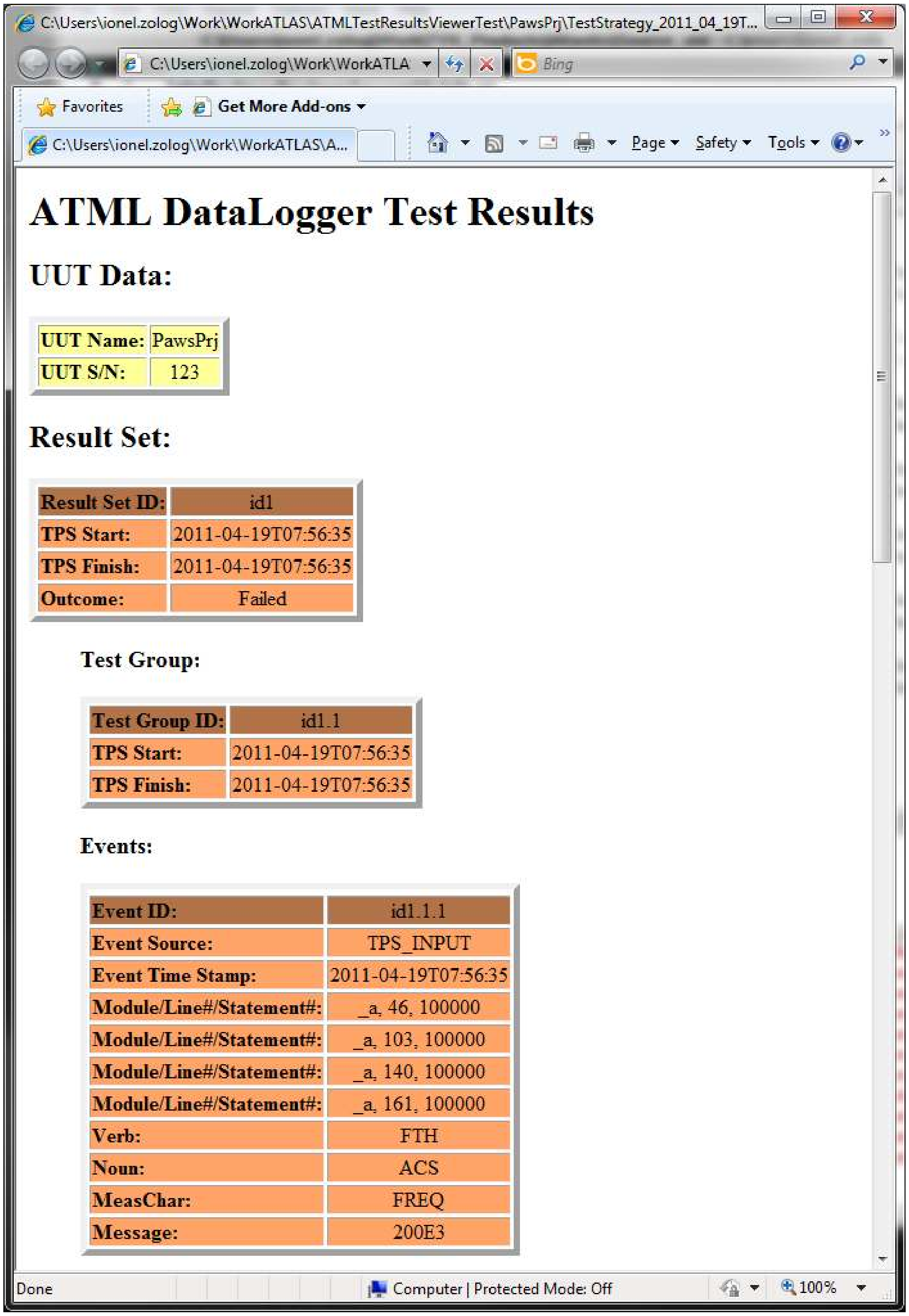
TerryColes
TonyBeckett
TracyMcQuillen
TYX_Products
UsrRelease
WorkATLAS
1641PawsDemo
ATMLTestResultsViewerTest
AutotestconFlowErr
BarteksTest
BarteksTest2
Beep
BillsPrj_PR11001
BugID504
BugID606
CassFile
CASSPCTest
CEMWizTest
CENTER_FREQ
ChangeReq750
ChrisATMLProj
ChristophesSigBasePrj
DeleteMeLater
Demo1641
DemoGRA_1
DemoGRA_2
DemoSigBase
DemoSigBase2
DepthOfModulation
DigitalInput

Name	Size	Created	Modified
PawsPrj	0 KB	4/19/2011 7:53 AM	4/19/2011 8:53 AM
TestStrategy_2011_04_19T07_56_12_000	0 KB	4/19/2011 7:56 AM	4/19/2011 10:04 AM
TestProcedureResult_2011_04_19T07_56_20_000.xml	9 KB	4/19/2011 7:56 AM	4/19/2011 7:56 AM
TestProcedureResult_2011_04_19T07_56_59_000.xml	1 KB	4/19/2011 7:57 AM	4/19/2011 7:57 AM
TestStrategy_2011_04_19T08_00_38_000	0 KB	4/19/2011 8:01 AM	4/19/2011 10:04 AM
TestProcedureResult_2011_04_19T08_00_41_000.xml	9 KB	4/19/2011 8:01 AM	4/19/2011 8:01 AM
TestProcedureResult_2011_04_19T08_01_18_000.xml	1 KB	4/19/2011 8:01 AM	4/19/2011 8:01 AM
TestStrategy_2011_04_19T08_01_36_000	0 KB	4/19/2011 8:01 AM	4/19/2011 10:04 AM
TestProcedureResult_2011_04_19T08_01_43_000.xml	9 KB	4/19/2011 8:01 AM	4/19/2011 8:01 AM
TestProcedureResult_2011_04_19T08_01_53_000.xml	1 KB	4/19/2011 8:01 AM	4/19/2011 8:01 AM
TestStrategy_2011_04_19T08_02_14_000	0 KB	4/19/2011 8:02 AM	4/19/2011 8:18 AM
TestProcedureResult_2011_04_19T08_02_20_000.xml	9 KB	4/19/2011 8:02 AM	4/19/2011 8:02 AM
TestProcedureResult_2011_04_19T08_02_25_000.xml	1 KB	4/19/2011 8:02 AM	4/19/2011 8:02 AM
TestStrategy_2011_04_19T08_42_15_000	0 KB	4/19/2011 8:42 AM	4/19/2011 8:42 AM
TestProcedureResult_2011_04_19T08_42_17_000.xml	9 KB	4/19/2011 8:42 AM	4/19/2011 8:42 AM
TestProcedureResult_2011_04_19T08_42_22_000.xml	1 KB	4/19/2011 8:42 AM	4/19/2011 8:42 AM
TestStrategy_2011_04_19T08_45_56_000	0 KB	4/19/2011 8:46 AM	4/19/2011 8:46 AM
TestProcedureResult_2011_04_19T08_45_59_000.xml	9 KB	4/19/2011 8:46 AM	4/19/2011 8:46 AM
TestProcedureResult_2011_04_19T08_46_04_000.xml	1 KB	4/19/2011 8:46 AM	4/19/2011 8:46 AM
TestStrategy_2011_04_19T08_46_49_000	0 KB	4/19/2011 8:47 AM	4/19/2011 9:19 AM
TestProcedureResult_2011_04_19T08_46_51_000.xml	9 KB	4/19/2011 8:47 AM	4/19/2011 8:47 AM
TestProcedureResult_2011_04_19T08_47_18_000.xml	1 KB	4/19/2011 8:47 AM	4/19/2011 8:47 AM

The user navigates in the left panel to the folder that contains Paws Project. Once the Load button is clicked, the viewer identifies all the test results contained as leaves of the loaded project. If they were created as a result of running a SigBase test strategy, they are grouped as in the picture above.

When the user clicks an xml file in the right panel, that file is translated to an intermediate html file named ATMLPawsTestResults.htm through an XSLT defined in <usr>\tyx\datalog\ATMLPawsTestResults.xsl. The intermediate html file is always placed as sibling of the xml input file.

In Microsoft Internet Explorer, the intermediate file looks like below.


C:\Users\ionel.zolog\Work\WorkATLAS\ATMLTestResultsViewerTest\PawsPrj\TestStrategy_2011_04_19T...
Favorites
Get More Add-ons
Page
Safety
Tools
Done
Computer | Protected Mode: Off
100%

ATML DataLogger Test Results

UUT Data:

UUT Name:	PawsPrj
UUT S/N:	123

Result Set:

Result Set ID:	id1
TPS Start:	2011-04-19T07:56:35
TPS Finish:	2011-04-19T07:56:35
Outcome:	Failed

Test Group:

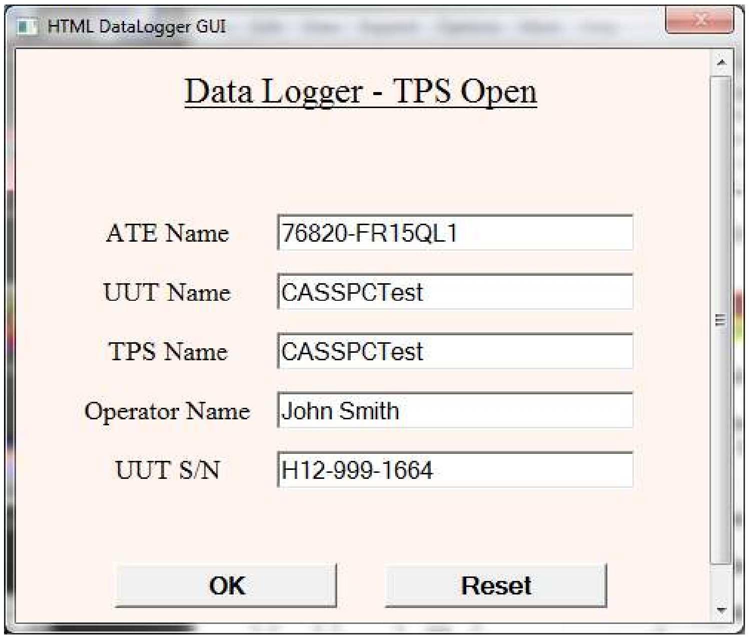
Test Group ID:	id1.1
TPS Start:	2011-04-19T07:56:35
TPS Finish:	2011-04-19T07:56:35

Events:

Event ID:	id1.1.1
Event Source:	TPS_INPUT
Event Time Stamp:	2011-04-19T07:56:35
Module/Line#/Statement#:	_a, 46, 100000
Module/Line#/Statement#:	_a, 103, 100000
Module/Line#/Statement#:	_a, 140, 100000
Module/Line#/Statement#:	_a, 161, 100000
Verb:	FTH
Noun:	ACS
MeasChar:	FREQ
Message:	200E3

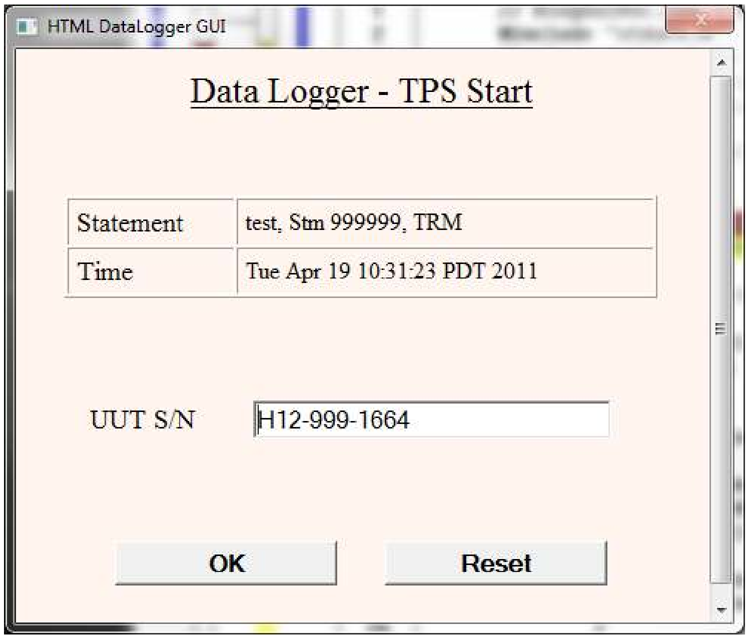
2.3.2 HTMLDlogGui.ini file info saved in registry

This file contained information about the default fields present in the <usr>\Tyx\Datalog\TPS_Open.htm and <usr>\Tyx\Datalog\TPS_Start.htm HTML datalogger input forms, as present in the pictures.



Data Logger - TPS Open

ATE Name	76820-FR15QL1
UUT Name	CASSPCTest
TPS Name	CASSPCTest
Operator Name	John Smith
UUT S/N	H12-999-1664



Data Logger - TPS Start

Statement	test, Stm 999999, TRM
Time	Tue Apr 19 10:31:23 PDT 2011

UUT S/N H12-999-1664

Starting with this release, this information is saved in registry in the HKEY_CURRENT_USER node. This way, even a user that is not an administrator has access to read or write this information.

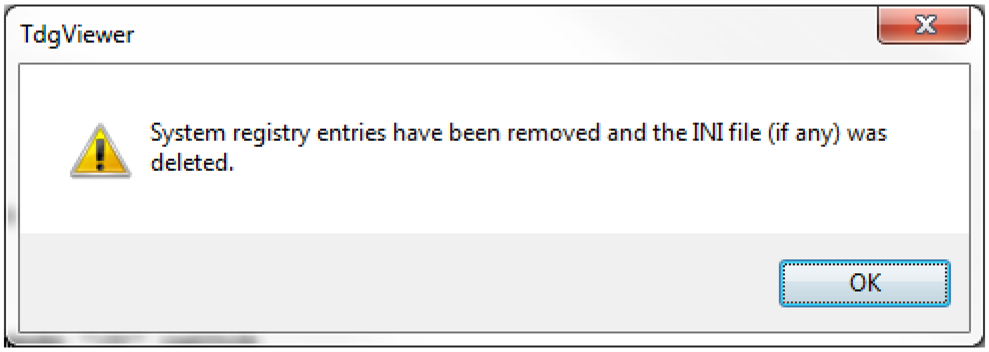
2.3.3 HaspWriterHL application creates universally distributed license files

The standard HaspWriterHL application is able to read and write universally distributed license files for the MATE subset. These license files only work along with our Alternate License Manager.

This application has under Tools pull-down menu two new entries: Read License File and Write License File. These entries help with the management of the new universally distributed license files.

2.3.4 TDGViewer does not report any messages at unregistering time (BugID 771)

At unregistering time, the TDGViewer used to pop the following message box:


TdgViewer

System registry entries have been removed and the INI file (if any) was deleted.

This message is not shown anymore.

2.4 Problem Reports

2.4.1 PR11019 PLI crashes when is retrieves tokens like “0.9E99” from an input file (BugID 741)

This problem has been diagnosed as a latent memory leakage problem for all the previous PLI.EXEs that we have released. This problem was only noticeable a MATE subset. The fix corrects the memory leakage and indirectly resolves the crash.

2.4.2 PR11026 TIME(0) ATLAS function does not return the correct time (BugID 762)

This ATLAS function is intended to return the number of seconds elapsed since 1-st of January 1970. The RTS has now a 64 bit integer to hold this integer.