TYX CORPORATION

Productivity Enhancement Systems

PAWS Studio Release Notes

Version 1.39.13
November 1, 2012

Table of contents

1 Paws Developer's Studio

Version 1.39.13

Release date: November 1, 2012

1.1 Critical Items

1.2 Known Limitations

1.3 Enhancements

1.4 Problem Reports

1.4.1 BugID 1100 PawsStudio Unable to Debug Atlas Modules Whose Names Are Longer Than 15 Characters

Our present compiler architecture supports atlas module names of maximum 15 characters in length, excluding the extensions. Our C:\usr\tyx\com\PawsDebugger.dll has been enhanced to mitigate this limitation. Paws Developer’s Studio loads this DLL when a Debug Session is started.

Unlike before, our Paws Studio Debugger and any RTS client that uses our Debugger RTS Adapter are able to manage Paws Projects that have atlas modules whose names are longer than 15 characters, as long as there are no distinctive atlas modules, belonging to the same Paws Project, and whose first 15 characters in their names are identical.

2 Run Time System

Version 1.39.13

Release date: November 1, 2012

2.1 Critical Items

2.2 Known Limitations

2.3 Enhancements

2.3.1 Our Primary Interop Assemblies Are Available in GAC and C:\usr\tyx\dotnet

When our Paws Studio software is installed, copies of all our Primary Interop Assemblies that go in Global Assembly Cache (GAC) are present in our <c:\usr>\tyx\dotnet folder. Before, our Primary Interop Assembles were only present in GAC.

2.4 Problem Reports

2.4.1 TaskID 1081: Provide User Information to our ATML Test Results Datalogger

This version of ATML Test Results datalogger implements the IXMLRecorderUserInformation interface, though which an RTS client can configure the ATE name, Operator Name, TPS Name, UUT Name, and UUT Serial Number, without using the standard GUI Html input forms. All these pieces of data will be filled in in the generated ATML Test Description document.

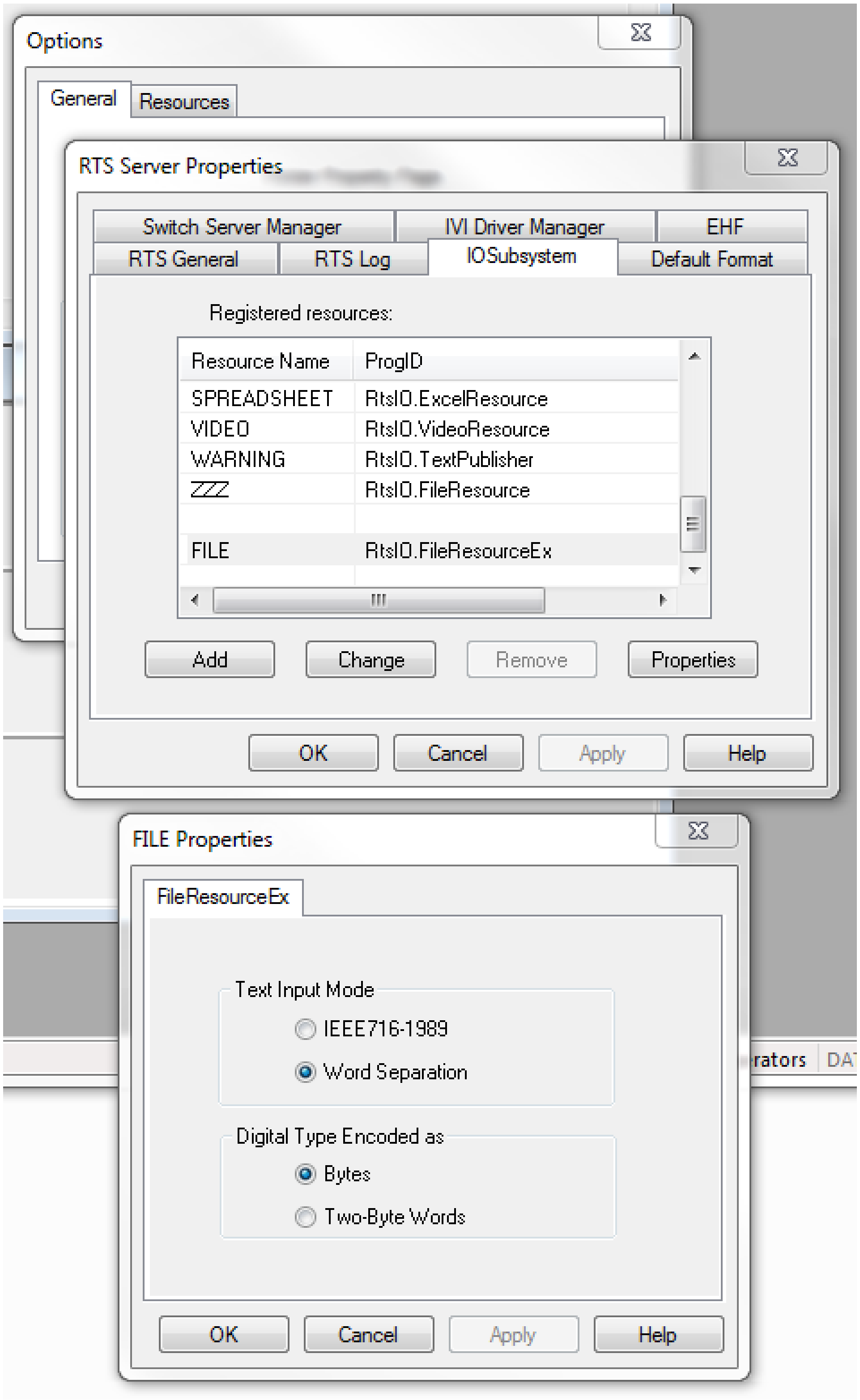
This is how the RTS client accesses this interface:

```
TYX.RtsAx.ITpsServer tpsServer =
(TYX.RtsAx.ITpsServer)this.tpsServerLiteClass;
TYX.RtsAx.IRtsSettings rtsSettings = tpsServer.Settings;

// check the default configuration
string sDefaultFormat = rtsSettings.DefaultFormat;

// enable the data logger
rtsSettings.Config = (int)TYX.RtsAx.RtsConfigOptions.OPT_CFG_PRODUCTION |
    (int)TYX.RtsAx.RtsConfigOptions.OPT_CFG_DATALOGGER;

// configure the desired datalogger recorder progID
rtsSettings.DlogRecProgID = @"TYX.ATMLTestResultsLogger.Recorder";

// get access to the datalogger subsystem (this is not the datalogger
recorder)
TYX.DataLogger.IDlogLogger dlogLogger =
(TYX.DataLogger.IDlogLogger)rtsSettings.DlogServer;

// configure the datalogger (please see the Data Logger tab in the RTS):
// Step 1 of 2; Operator Input Events group (note:
// have this equal with zero, if no GUI input is required from the user):
```

```
dlogLogger.GuiEvents =
(int)TYX.DataLogger.DlogEventMasks.DLOG_EVENT_TPS_OPEN_MASK |
(int)TYX.DataLogger.DlogEventMasks.DLOG_EVENT_TPS_CLOSE_MASK;

// Step 2 of 2; Record group:
dlogLogger.Options = dlogLogger.Options |
(int)TYX.DataLogger.DlogOptions.DLOG_LOG_BUS;

// finally, get access to the data logger recorder and its configuration
// the type of this object must be consistent with the previously
// configured recorder progID
TYX.ATMLTestResultsLogger.IXMLRecorderGeneralOptions recorder =
(TYX.ATMLTestResultsLogger.IXMLRecorderGeneralOptions)rtsSettings.DlogRecorder
;

// extract some configuration of the recorder
string s = recorder.XMLFilePrefix;

// this would be the place to configure
// through software the following information:
// UUT_SN, UUT_NAME, ATE_NAME, TPS_NAME, OPERATOR_NAME.
// This information gets logged in the ATML Test Results document during
// the TPS_OPEN and TPS_START events
TYX.ATMLTestResultsLogger.IXMLRecorderUserInformation userInformation =
(TYX.ATMLTestResultsLogger.IXMLRecorderUserInformation)rtsSettings.DlogRecorde
r;

userInformation.ATEName = "ATEName123";
userInformation.OperatorName = "OperatorName123";
userInformation.TPSName = "TPSName123";
userInformation.UUTName = "UUTName123";
userInformation.UUTSerialNumber = "UUTSerialNumber123";
```

2.4.2 TaskID 1086: Provide an IEEE716.89 FILE Resource That Has Digital Type Encoded as Bytes

`RtsIO.FileResourceEx` provides as alternative to `RtsIO.FileResource`, a FILE that encodes the digital data as bytes, not as words. In order to benefit of this resource, the RTS | Control | Options | RTS Property Pages | IO Subsystem tab has to define as FILE the `RtsIO.FileResourceEx` ProgID. Its settings (click Properties) look like:



Registered resources:

Resource Name	ProgID
SPREADSHEET	RtsIO.ExcelResource
VIDEO	RtsIO.VideoResource
WARNING	RtsIO.TextPublisher
ZZZ	RtsIO.FileResource
FILE	RtsIO.FileResourceEx

Text Input Mode

○ IEEE716-1989

● Word Separation

Digital Type Encoded as

● Bytes

○ Two-Byte Words

Besides the Digital Type encoding feature, unlike `RtsIO.FileResource`, our new `RtsIO.FileResourceEx` does not expect or generate a header in its physical file on disk.

2.4.3 BugID 1085: Unable to Print a Text in a Rectangle of Stars

The essence of this bug was that our standard PRINTER could not print text in a rectangle of stars, when both the text and stars were output through FORTRAN format. In the atlas sample below, the stars are now printed at columns 8 and 73:

```
000001 BEGIN, ATLAS PROGRAM 'IO-BENCH' $

     10 REQUIRE, 'CRT', I-O DEVICE,
           CAPABILITY,
           LINE-LENGTH 80 CHAR,
           PAGE-SIZE 24 LINES $

 000100 DECLARE, DECIMAL, STORE, 'X', 'Y', 'Z'            $
E Test number 1 $
 070000 CALCULATE, 'X' = 555.5555555555555555555 $
        OUTPUT, USING 'PRINTER', USING 'CRT',
     ((8X,'******************************************************************'))
     ((8X,'*',T73,'*',T11,'(E20.1)   ',E20.1)  'X' )
     ((8X,'*',T73,'*',T11,'(E20.2)   ',E20.2)  'X' )
     ((8X,'******************************************************************'))
     $
E999999 TERMINATE, ATLAS PROGRAM 'IO-BENCH' $
```

Now, the output PRINTER file looks like:

```
        ******************************************************************
        * (E20.1)               0.6E+03                                  *
        * (E20.2)              0.56E+03                                  *
        ******************************************************************
```

2.4.4 BugID 1080: Added support for INPUT statements with FORTRAN specification BN and BZ

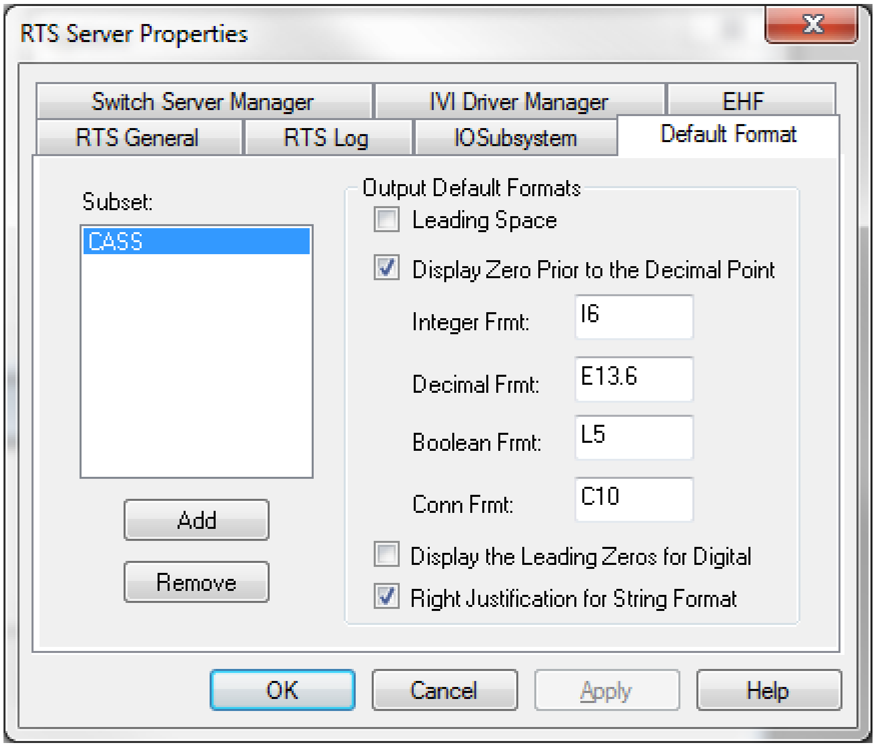
The following atlas code works correctly, in conjunction with a `RtsIOEx.RecordOrientedFileResource` FILE configured in the IOSubsystem tab.

```
000001 BEGIN, ATLAS PROGRAM 'INPUT WITH BN AND BZ' $
     10 REQUIRE, 'CRT', I-O DEVICE,
           CAPABILITY,
           LINE-LENGTH 80 CHAR,
           PAGE-SIZE 24 LINES $
     20 REQUIRE, 'WFILE1', I-O FILE,
           CAPABILITY,
           FILE-ORGANIZATION RELATIVE,
           FILE-FORM FORMATTED,
           RECORD-TYPE VARIABLE,
           RECORD-LENGTH 80 BYTES $
 000100 DECLARE, MSGCHAR, STORE, 'T-24', 24 CHAR $
        DECLARE, INTEGER, STORE, 'K', 'L'             $
E Test number 1 $
 100000 ENABLE, FILE, NEW 'WFILE1', FILE-ACCESS DIRECT $
        CALCULATE, 'K' = 0 $
        CALCULATE, 'T-24' = C'123' $
        OUTPUT, USING 'WFILE1', RECORD-NUMBER 7,((A5) 'T-24') $
 100010 INPUT,  USING 'WFILE1', RECORD-NUMBER 7,(I1, BN, I5) 'L', 'K' $
        DISABLE, FILE, 'WFILE1' $
E999999 TERMINATE, ATLAS PROGRAM 'INPUT WITH BN AND BZ' $
```

The INPUT statement at 100010 may contain the `BN` or `BZ` FORTRAN specification.

2.4.5 BugID 1083: "A" FORTRAN Specification May Enforce Right Justification

The right or left justification for strings that are output is configurable through RTS | Control | Options | RTS Property Pages | Default Format:



RTS Server Properties

Subset: CASS

Output Default Formats

☐ Leading Space

☑ Display Zero Prior to the Decimal Point

Integer Frmt: I6

Decimal Frmt: E13.6

Boolean Frmt: L5

Conn Frmt: C10

☐ Display the Leading Zeros for Digital

☑ Right Justification for String Format

In the tab above, Right Justification is chosen.

This version also implements Right Justification for the strings that are input, through a `RtsIOEx.RecordOrientedFileResource` FILE.

2.4.6 TaskID 1089: All ATLAS Variables Are Reset RTS Executes FINISH and TERMINATE Statements

For all CASS stations, our RTS initializes all TPS variables with their default values, when it executes the FINISH and TERMINATE ATLAS statements. As an outcome of this change, the TPS variables get initialized with their default values when the user clicks the Reset button, too. It is known that a user click of the Reset button makes the RTS interrupt its current TPS execution and jump to the TERMINATE statement.

2.4.7 TaskID 1084: RTS Does Not Display the Leading Zero If There Is No Room for It

When the RTS is configured to display the leading zero, and it runs under a CASS station, the RTS does not display the leading zero if there is no room for it.

The following atlas:

```
000001 BEGIN, ATLAS PROGRAM 'IO-LEADINGZERO' $
     10 REQUIRE, 'CRT', I-O DEVICE,
           CAPABILITY,
           LINE-LENGTH 80 CHAR,
           PAGE-SIZE 24 LINES $
 000100 DECLARE, DECIMAL, STORE, 'X', 'Y', 'Z'            $
E Test number 1 $
 070000 CALCULATE, 'X' = 555.5555555555555555555 $
```

```
        OUTPUT, USING 'CRT',
     ((8X,'*',T73,'*',T11,'(E20.15) ',E20.15) 'X' )
     ((8X,'*****************************************************************'))
     $
E999999 TERMINATE, ATLAS PROGRAM 'IO-LEADINGZERO' $
```

used to produce the following output:

```
Binary files "testStarMngm", version 20121010
Date stamp: Mon Oct 29 16:03:49 2012
Built In LEX Information
        * (E20.15) ********************                                 *
        *****************************************************************
```

due to the fact that the normal encoding for E20.15 for ‘X’ would have been

```
0.555555555555556E+03
```

that requires 21 characters, while one 20 are explicitly specified.

Now, with the policy of not displaying the leading zero in effect, the output will be:

```
Binary files "testStarMngm", version 20121010
Date stamp: Mon Oct 29 16:10:15 2012
Built In LEX Information
        * (E20.15) .555555555555556E+03                                 *
        *****************************************************************
```