

TYX CORPORATION

Productivity Enhancement Systems

---



# PAWS Studio Release Notes

Version 1.35.7  
October 12, 2007

# Table of contents

<b>1</b>	<b>Paws Developer's Studio</b>	<b>3</b>
1.1	Critical Items	3
1.2	Known Limitations	3
1.3	Enhancements	3
1.3.1	New XML format for EHF – Enhanced History File [Displayed in PAWS Developer Studio]	3
1.3.2	CRC information in built files	4
1.3.3	PawsInfo.exe shows new CRC information for TYX compiler generated binary files	4
1.4	Problem Reports	5
1.4.1	PR 07071	5
<b>2</b>	<b>Run Time System</b>	<b>6</b>
2.1	Critical Items	6
2.2	Known Limitations	6
2.3	Enhancements	6
2.3.1	New XML format for EHF – Enhanced History File	6
2.4	Problem Reports	9
2.4.1	PR 07059	9
2.4.2	PR 07069	9
2.4.3	PR 07019	9
2.4.4	PR 07060	10

# 1 Paws Developer's Studio



Version 1.35.7

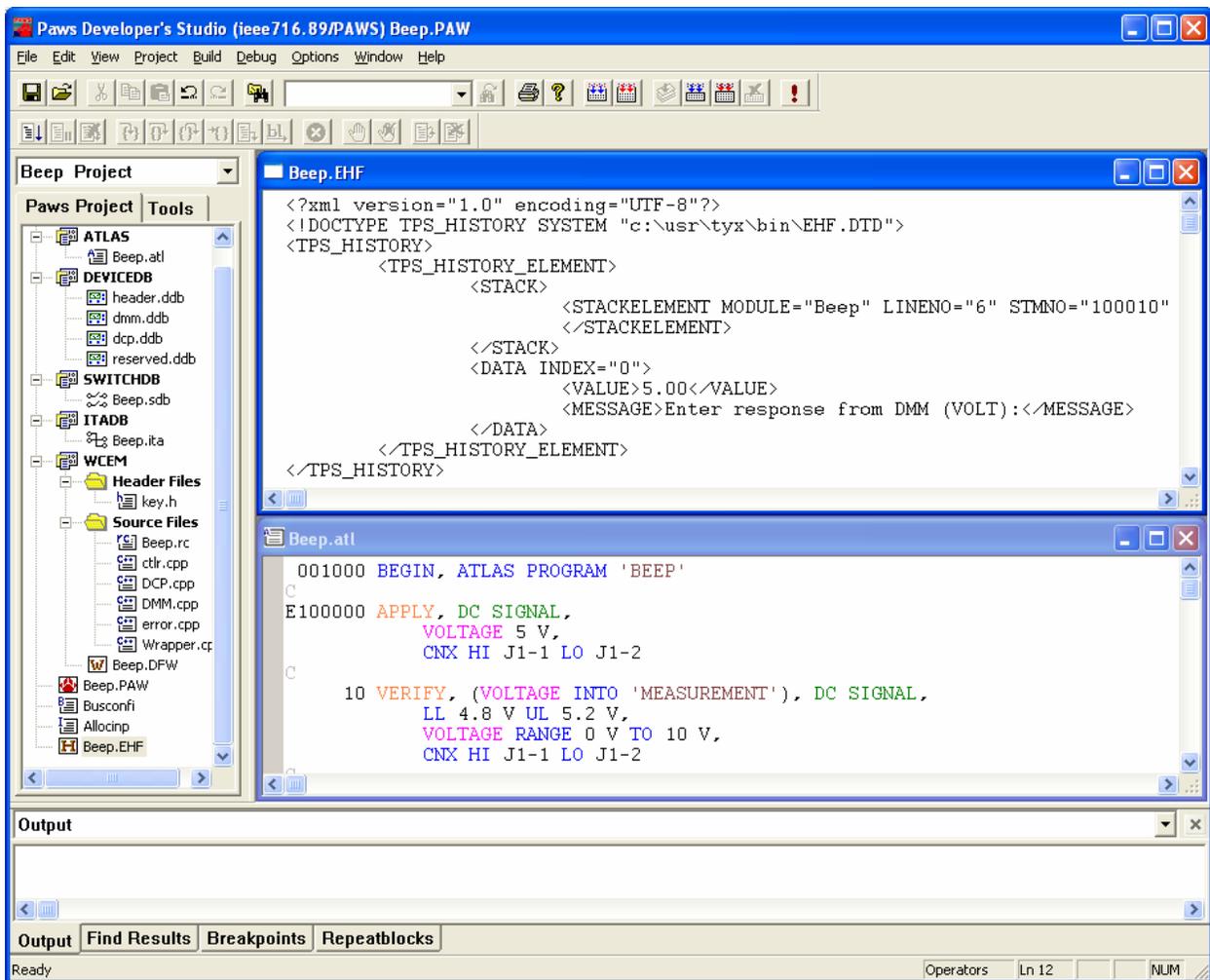
Release date: October 12, 2007

## 1.1 Critical Items

## 1.2 Known Limitations

## 1.3 Enhancements

### 1.3.1 New XML format for EHF – Enhanced History File [Displayed in PAWS Developer Studio]



Paws Developer Studio will display the new Enhanced History File (EHF) - XML document from the Paws Project Tree, using the double click or by selecting the EHF and using the Context menu "Open" option.

Users may also edit the file to update simulated values or add new <TPS\_HISTORY\_ELEMENT> elements to simulate new MEASUREMENTS. More information provided in the **Run Time System** section of this document.

### 1.3.2 CRC information in built files

In order to have a consistent project, all files within the project have to use the same version of the LEX files. This is easily done when all files in the project are sources files and the user performs a "rebuild all". Problems are likely to occur if some files in a project are built with one version of a LEX file and other with another version of a LEX file and then all combined in a project.

This can happen when you include binary files in your project (OBJ, DAT, DEV, SWX), or you make changes to the LEX files and then a "build" as opposed to a "rebuild all". Such problem can be very time consuming to troubleshoot for users.

In order to avoid such problems, Cyclical Redundancy Check (CRC) information has been added to the building of the LEX files.

Note: Changes that 2 versions of the LEX files have the same CRC number is extremely unlikely but possible.

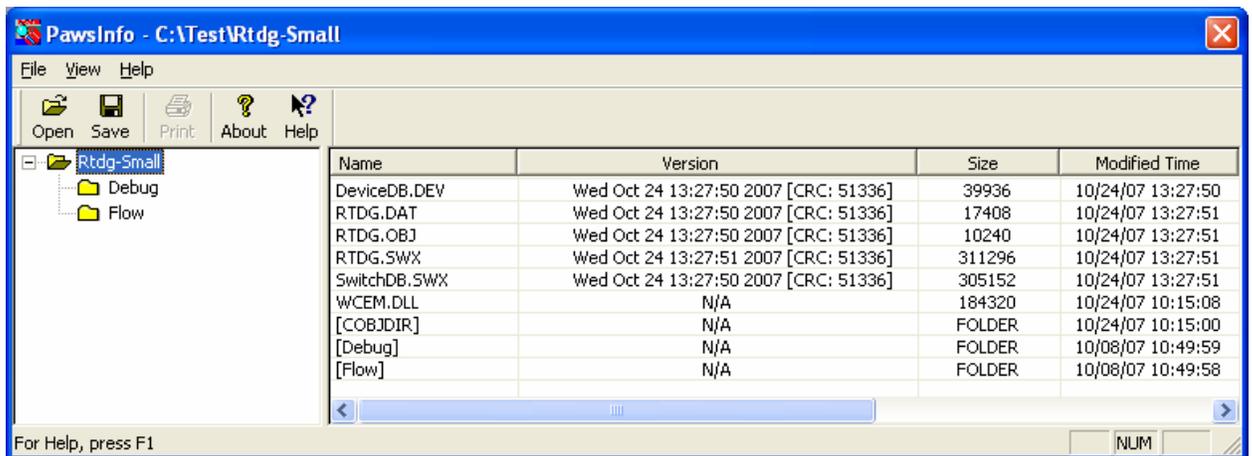
As a result, all binaries that are built with the new version of the LEX file that includes the CRC information will also include that CRC information and the Paws Studio will check for consistency and issue an error if you are using binary files that have a CRC information number different from the LEX file that you are using. Paws Developer Studio would force rebuild of a project or incompatible modules if such an error is detected.

Users can easily verify the current CRC information number of the LEX file by using PawsInfo.exe application. PawsInfo.exe can also give you the CRC information number for the OBJ, DAT, DEV and SWX binary files.

For backward compatibility old LEX, .DAT, .OBJ, .SIG, .DEV and .SWX files may not have a CRC value, unless PAWS users recompile their LEX files. To take advantage of this Paws Developer Studio, compiler support and Run Time System capability, and to avoid allocation/loading pitfalls due to incompatibility, TYX encourages its users to rebuild their LEX files to generate a CRC information number within the binary file (LexDB.lex) and subsequently rebuild their project to incorporate the CRC number in PAWS Studio generated OBJ, DAT, DEV and SWX binary files.

### 1.3.3 PawsInfo.exe shows new CRC information for TYX compiler generated binary files

To support users in eliminating allocation and loading issues, PawsInfo.exe application will display the CRC value used to generate various binary files like ... LEX, .DAT, .OBJ, .SIG, .DEV and .SWX. Users may examine the CRC number to verify that the associated files are compatible with their LEX DB.



## **1.4 Problem Reports**

### **1.4.1 PR 07071**

Paws Developer Studio was unable to set breakpoints in Debug mode. This problem was detected internally shortly after Release 1.35.6 and customers using this version with compilers other than CASS were requested to roll back to version 1.35.4 to use this feature. Subsequently, this problem has been fixed with this Release and all Debug related features are enabled.

## 2 Run Time System



**Version 1.35.7**

**Release date: October 12, 2007**

### 2.1 Critical Items

### 2.2 Known Limitations

### 2.3 Enhancements

#### 2.3.1 New XML format for EHF – Enhanced History File

Enhanced history file information is now stored in XML format.

The advantages of this are:

- Users can easily view simulated values in the XML Enhanced History File (EHF) which can be opened in Paws Studio Developer editor.
- Users can edit the XML document to update, correct or add new simulated values which would be displayed by default for the next RTS simulation run. TYX distributes the XML document type definition and XML schema files which can be used to determine the XML structure and validate an EHF XML document.
- Users can configure a “Match Criteria” which the RTS should use to detect a match for a given ATLAS measurement and a simulated default value provided in the EHF XML document.
- Backward compatibility. Binary EHF files generated and used by older versions of RTS can be converted to the EHF XML format.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TPS_HISTORY SYSTEM "c:\usr\tyx\bin\EHF.DTD">
<TPS_HISTORY>
  <TPS_HISTORY_ELEMENT>
    <STACK>
      <STACKELEMENT MODULE="Rtdg" LINENO="124" STMNO="300080" VLC="5571">
      </STACKELEMENT>
    </STACK>
    <DATA INDEX="0">
      <VALUE>100</VALUE>
      <MESSAGE>Enter response from DSP (TIME):</MESSAGE>
    </DATA>
  </TPS_HISTORY_ELEMENT>
  <TPS_HISTORY_ELEMENT>
    <STACK>
      <STACKELEMENT MODULE="Rtdg" LINENO="148" STMNO="400030" VLC="6151">
      </STACKELEMENT>
    </STACK>
    <DATA INDEX="0">
      <VALUE>10 11 12 13</VALUE>
      <MESSAGE>Enter response from DWG (VALU [4]), 100 bytes :</MESSAGE>
    </DATA>
  </TPS_HISTORY_ELEMENT>

```

Figure 1

Above is a sample snippet of an EHF file displayed by Paws Developer Studio.

It shows the hierarchical structure of the EHF. This conforms to the EHF.dtd [Document type definitions] and EHF.xsd [XML Schema Definitions] distributed by TYX Corporation and referenced by the EHF XML document.

```

<TPS_HISTORY_ELEMENT>
  <STACK>
    <STACKELEMENT MODULE="Rtdg_Small" LINENO="12" STMNO="2040" VLC="3405">
    </STACKELEMENT>
    <STACKELEMENT MODULE="Rtdg_Small" LINENO="39" STMNO="100070">
    </STACKELEMENT>
  </STACK>
  <DATA INDEX="0">
    <VALUE>1</VALUE>
    <MESSAGE>Enter response from DTC (FREQ):</MESSAGE>
  </DATA>
  <DATA INDEX="1">
    <VALUE></VALUE>
    <MESSAGE>Enter response from DTC (FREQ):</MESSAGE>
  </DATA>
  <DATA INDEX="2">
    <VALUE>3</VALUE>
    <MESSAGE>Enter response from DTC (FREQ):</MESSAGE>
  </DATA>
</TPS_HISTORY_ELEMENT>

```

Figure 2

Noticeable from the XML displayed in *Figure 2* is the <STACK> element which is used to perform the “Match” for a measurement requested at a particular ATLAS instruction (Atlas Module, Atlas Statement Number (optional), Atlas Line Number). The <STACK> element above describes a simulated value “1” for a MEASUREMENT performed in the “Rtdg\_Small.atl” module at Line number 12. This ideally describes the “top” of the stack element. As presented in the XML above the <STACK> shows multiple <STACKELEMENTS>. Other stack values may be provided if the MEASUREMENT lies within an ATLAS PROCEDURE and the PROCEDURE is called from the ATLAS main program. (as in the case above ... where the above MEASUREMENT is called from Line number 39 in the main ATLAS program.

Multiple <DATA> elements with incremental “INDEX” attribute depict the MEASUREMENT is called multiple times from the given <STACK> location. [In the case above, the PROCEDURE was called 3 times in a loop and the user choose to supply a default value “1” the first time, blank value when called again and the value “3” when called yet again ...and so on]

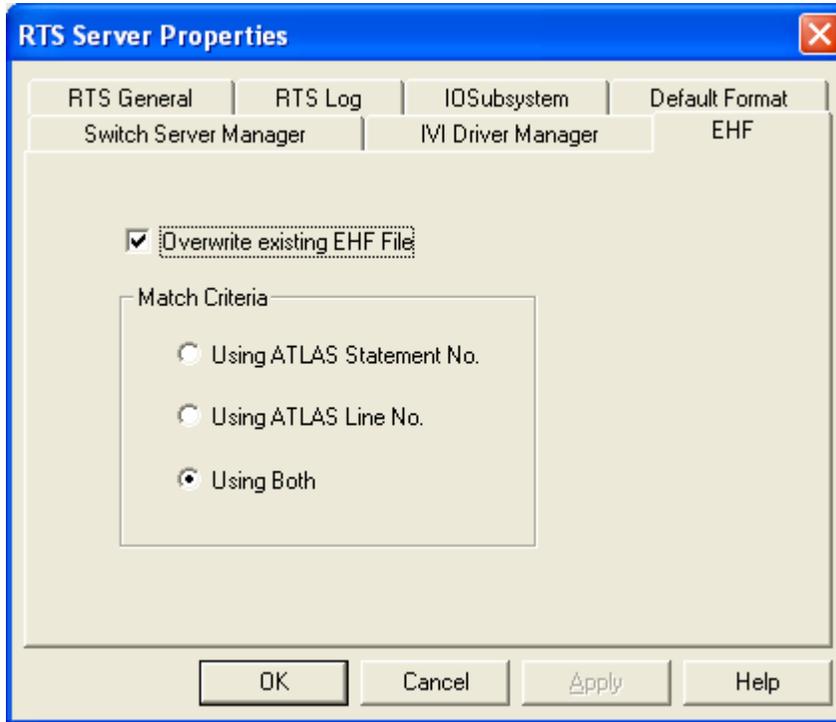


Figure 3

EHF settings can be configured using the EHF Tab in the RTS Server Properties page. Click “RTS→Options” menu and select the “RTS Property Pages” button.

User may choose to overwrite his existing EHF file for every ATLAS run, or uncheck this so that his initial EHF defaults are not tampered with. Users may also provide a “Match Criteria” which the RTS would use for creating a MATCH. In case ATLAS Statement numbers are not available in a file, RTS could be instructed to perform a match based on Line numbers and Module Name only. The Module Name is always a required parameter and is known to the RTS. The “VLC” attribute of a “STACKELEMENT” may be updated by the RTS and need not be provided by the user.



If an EHF file does not exist for an ATLAS project and an RTS simulation test is run where a MEASUREMENT is encountered, the RTS Measurement Dialog will display empty. New simulation values provided by the user will be stored in the EHF XML document. For this first run, user may see the following “Warning” message describing that an old EHF file did not exist. This is normal and the warning will not be seen for subsequent runs of the ATLAS project.

## 2.4 Problem Reports

### 2.4.1 [PR 07059](#)

This problem report has been fixed by the CRC enhancement. Run Time System will fail trying to load inconsistent binary files (i.e. binary files generated with different versions of LEX), provided the LEX was rebuilt using this Release of the Run Time System.

### 2.4.2 [PR 07069](#)

This problem report has been fixed in lieu of the changes made to PawsInfo.exe.

### 2.4.3 [PR 07019](#)

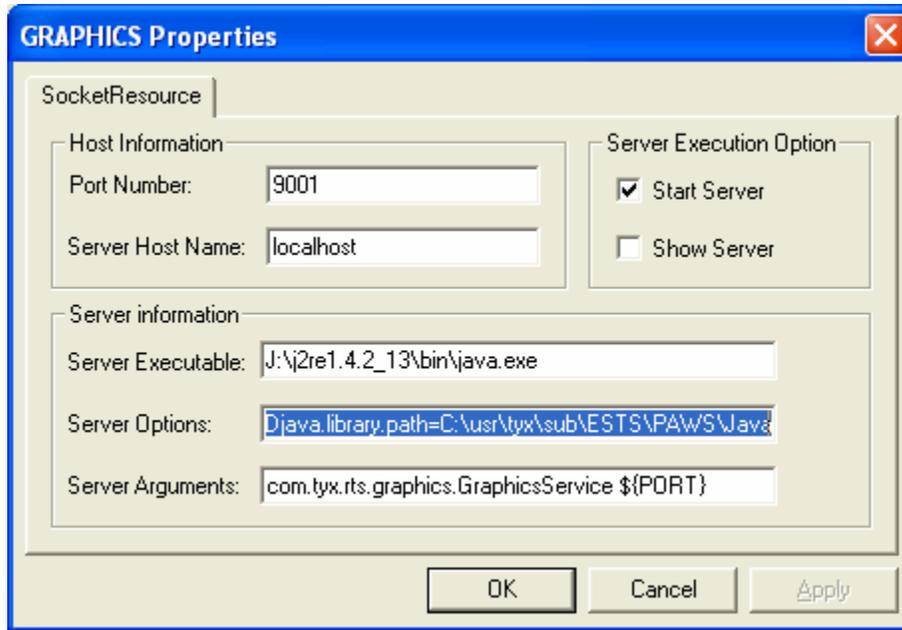
The problem report required an enhancement to be provided in the Graphics Viewer application. “ESTS/IFTE” subset users will now be able to use a “CTRL+R” [short cut key combination] from the Graphics Viewer application to send a “Reset” to the WRTS. To handle this translation TYX provide a new native “RtsControl.dll”. This dynamic link library will be provided by TYX Corporation, and will reside in the folder “<usr>\tyx\sub\ESTS\IFTE\PAWS\Java”.

To avail of this enhancement capability in the Graphics Viewer application, ESTS/IFTE users are requested to update the properties of the “**GRAPHICS**” resource.

In the WRTS application, select “Control”→“Options”. Click on “RTS Property Pages” and select the “GRAPHICS” resource and click on “Properties”. On the GRAPHICS Properties screen, **append** the following text to the “Server Options” text box.

**-Djava.library.path=<usr>\tyx\sub\ESTS\PAWS\Java**

Note: <usr> should be replaced with the directory PAWS is installed on the system. In the screen below its “C:\usr”.



The key “java.library.path” provides the Java Runtime with a path to find the “RtsControl.dll” native library.

Finally click “OK” on all the dialogs and restart the run time system. The WRTS will now be configured to receive the “CTRL+R” [short cut key] message and perform a “RESET”.

#### 2.4.4 PR 07060

The problem report fixes font size used for display in the Graphics Viewer application, for Menu Items. New version 3.10 of the application has been provided with this release.