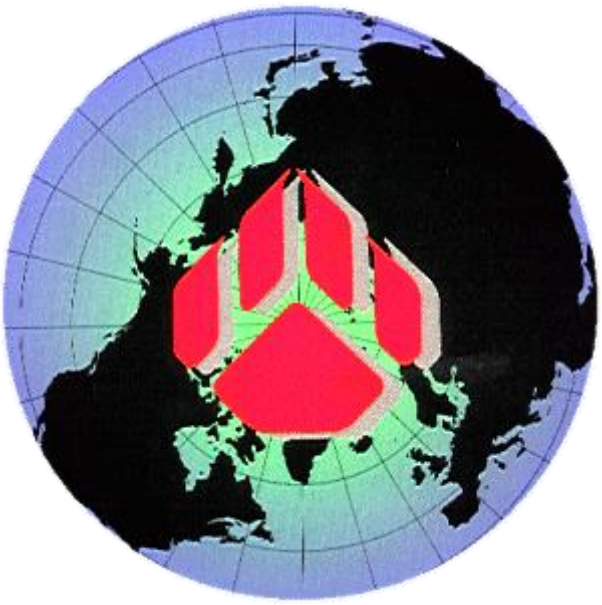
TYX CORPORATION

Productivity Enhancement Systems



PAWS Studio Release Notes

Version 1.39.10
July 12, 2012

Table of contents

1 Paws Developer's Studio

Version 1.39.10

Release date: July 12, 2012

1.1 Critical Items

1.2 Known Limitations

1.3 Enhancements

1.3.1 1641 Carrier Language translator into atlas implements the MessageBox.Show method

The new version of 1641 Carrier Language translator into atlas allows the compilation of the following carrier language code:

```
class A
{
  public void main()
  {
    DialogResult a = MessageBox.Show("MyTextToDisplay", "MyCaption",
                                     MessageBoxButtons.OKCancel);
    if (a == DialogResult.OK)
      // you may do something else here
      Console.WriteLine("OK");
    else if (a == DialogResult.Cancel)
      // you may do something else here
      Console.WriteLine("Cancel");
    else
      // you may do something else here
      Console.WriteLine("Unknown");
  }
}
```

This enhancement emulates the functionality of the `Show` method of the `System.Windows.Forms.MessageBox` C# class.

1.3.2 Paws Studio is able to identify errors and warnings of the preprocessor step

This version of Paws Studio is able to recognize warnings and errors at the preprocessor stage, and report the error or warning place in the input file of the preprocessor stage. This is done if the user double clicks the preprocessor error or warning text in the Output Window.

1.3.3 Paws Studio sends to the preprocessor the LU files and the allocation file

This version of Paws Studio allows a preprocessing of the LU files or the allocation input file, before they go to the ITA compiler or the atlas allocator.

1.3.4 Code optimized to avoid the random access violation error at project building time

This version of Paws Studio has its code optimized, so it limits the occurrence of the following type of random building error:

```
Encountered a sharing violation while accessing C:\usr\paws\sample\TSP.LNK.
Execution complete - 1 error(s), 15 warning(s)
```

1.4 Problem Reports

2 Run Time System

Version 1.39.10

Release date: July 12, 2012

2.1 Critical Items

2.2 Known Limitations

2.3 Enhancements

2.3.1 RTS C++ code compiled with /EHa instead of /EHsc

RTS is expected to treat in a more reliable way the customer's code present in User BIFs, Input/Output Resources, COM NAMs and CEMs. When unhandled exceptions are thrown from the customer's code, the RTS is expected to catch them and end the TPS gracefully instead of simply crashing.

2.4 Problem Reports

2.4.1 PR12004 (BugID 4) Unchecked Interrupts While RTS Executes an INPUT Statement – solution reviewed

The RTS version 1.39.9 attempted a fix of this issue. In its release notes, the following text reports its resolution:

> *This problem report underlines that the previous RTS had not checked for interrupts while it executed an INPUT statement. The present RTS has been modified to check every 10 seconds whether an interrupt has been issued, while it waits for the user to complete an INPUT statement, in an 85 based ATLAS flavor. This approach has been implemented for all dedicated INPUT resources that TYX provide (including Java Graphics Viewer). However, this mechanism cannot work for customized Single Threaded Apartment COM input resources that do not implement a wait for user while serving the message pump of the current thread.*

The fix released with 1.39.9 aborts an input dialog popped as a result of executing an 85 style INPUT, each time SetFault is called with an error code, regardless of whether the interrupt procedure of that error code is enabled or disabled.

The fix released with 1.39.10 aborts an input dialog popped as a result of executing an 85 style INPUT AIL, if only SetFault is called with an error code and that error code points to an enabled interrupt procedure.

The check for the interrupts during an 85 style INPUT has been also shortened from 10 seconds to 1 second.