TYX CORPORATION

Productivity Enhancement Systems

PAWS Studio
Release Notes

Version 1.37.12
September 3, 2009

Table of contents

1   Paws Developer's Studio

Version 1.37.12
Release date: September 3, 2009

1.1   Critical Items

1.2   Known Limitations

1.2.1   Limitations Regarding the Intellisense feature

1) The Intellisense module described at 1.3.1 uses a different syntax definition set of BNF productions and a different parsing engine model than the atlas compiler Paws Studio traditionally uses to compile the atlas files and build their object files. As a result of this, the proposed tokens offered by the Intellisense module might not be accepted by the atlas compiler at compile time.

2) The user has to be aware that the proposed tokens offered as continuation solution at atlas editing time by the Intellisense module are not assessed whether or not they are semantically correct. As an example, an atlas variable name as a proposed next token in a CALCULATE statement is not validated through a previous check on its definition.
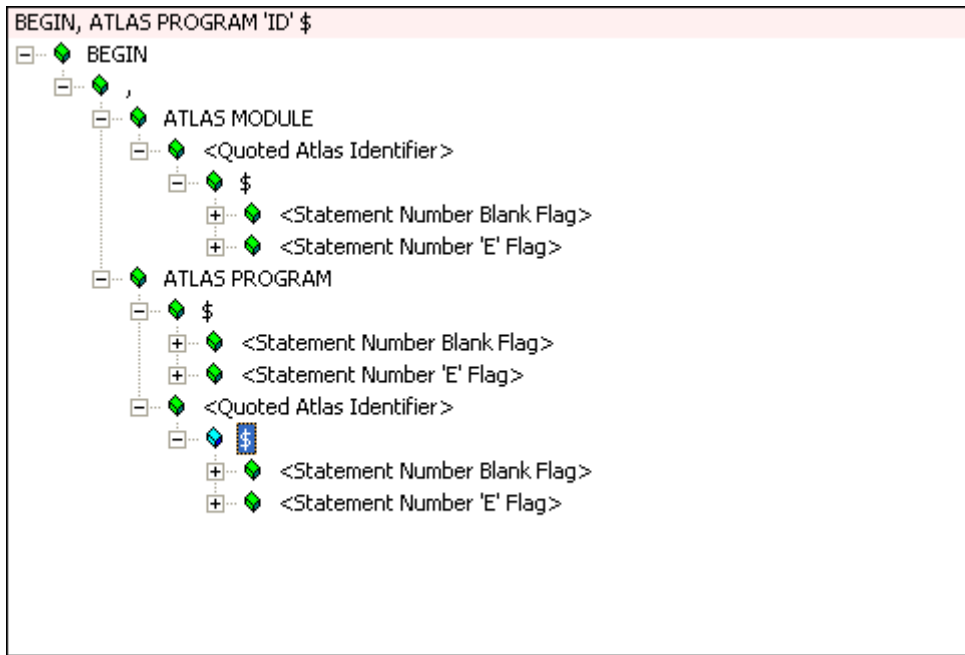
1.3   Enhancements

1.3.1   Intellisense help for the atlas subsets IEEE716.89 and CASS

This version of PAWS Developer Studio has been enhanced to offer intellisense help at atlas editing time for two subsets. In the future, most subsets will have this feature available.

The intelllisense feature gives the user as choices the next valid atlas tokens at the current editing point in the user's atlas file. The proposed atlas tokens are based on the atlas language syntax definition of the current subset and the lexical database content (the lexDB) of the current station. They are calculated by performing a partial parsing of the atlas file from the previous atlas statement until to the point where the editing caret is.

Once the PAWS project is loaded, a full parsing of all the project atlas files is performed. This is always done in a low priority second thread of PAWS Studio, so it does not impede on the normal responsiveness of the Studio application. During the initial parsing step, the parser internal state is saved at the termination of each atlas statement. As mentioned above, this speeds up the process of getting the proposed tokens due to the fact that the up-to-the-editing-point parsing is done starting from the previous atlas statement, not from the beginning of the file.

By default, the intellisense dialog is invoked by clicking Ctrl+F1. At the beginning of the atlas file, this dialog looks like the following.

```
BEGIN, ATLAS PROGRAM 'ID' $
⊟⬧ BEGIN
  ⊟⬧ ,
    ⊟⬧ ATLAS MODULE
      ⊟⬧ <Quoted Atlas Identifier>
        ⊟⬧ $
          ⊞⬧ <Statement Number Blank Flag>
          ⊞⬧ <Statement Number 'E' Flag>
    ⊟⬧ ATLAS PROGRAM
      ⊟⬧ $
        ⊞⬧ <Statement Number Blank Flag>
        ⊞⬧ <Statement Number 'E' Flag>
      ⊟⬧ <Quoted Atlas Identifier>
        ⊟⬧ $
          ⊞⬧ <Statement Number Blank Flag>
          ⊞⬧ <Statement Number 'E' Flag>
```

The user can navigate through the proposed tokens in the intellisense dialog. The edit control at the top of the dialog shows the valid continuation of the atlas statement at the point where the editing caret is. Once the intellisense dialog pops up, the user either clicks Enter and the content of the edit control at the top of the dialog goes into the edited file, or he clicks Escape and the proposed continuation is disregarded. In both cases the intellisense dialog closes.

The intellisense module is lexdb sensitive. For example, if the intellisense is invoked after the following incomplete atlas statement

```
100040 SETUP, (AC-COMP
```

the choices for nouns are selected based on whether or not they own as modifier the already available measure characteristic. In the case above, only DC SIGNAL noun admits AC-COMP as modifier.
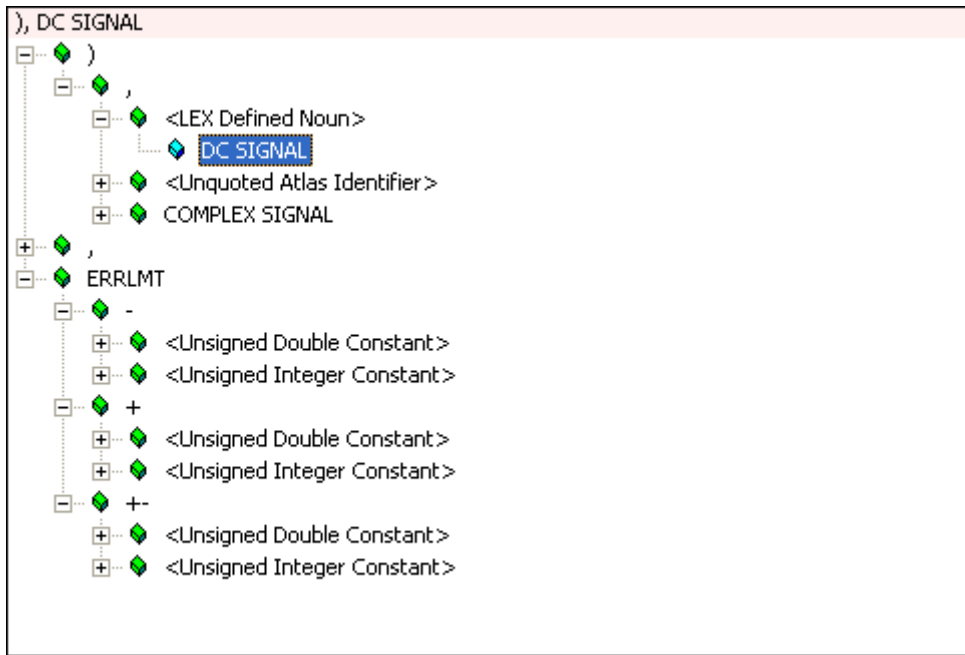
Besides nouns and modifiers, all the dimensions are also lexDB sensitive. If the above measure characteristic field contains the ERRLMT subfield, the intelisense tool offers as choices, based on lexDB information, only the dimensions that are appropriate for the current modifier:
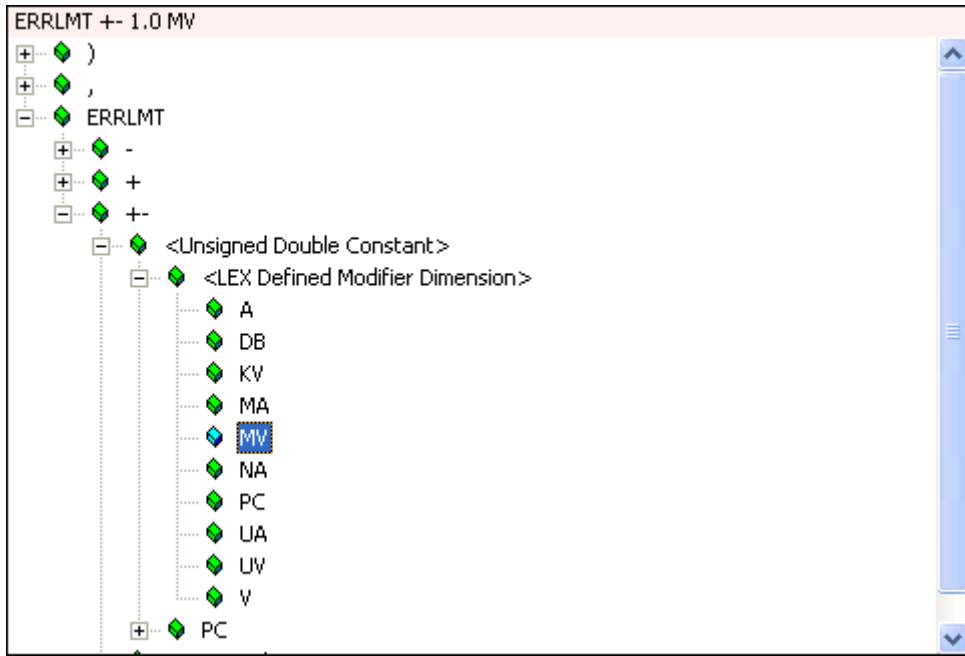
The user has control over the activation key for the intellisense dialog. By default this is Ctrl+F1. If this needs to be changed, the user has to configure the property page present at Options\Studio:
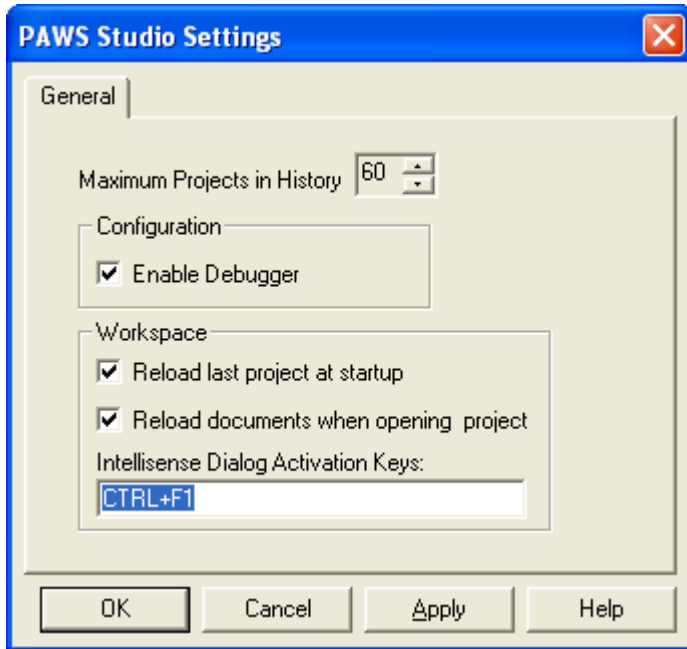
Important notes about the intellisense module:

When the Paws Studio is turned on, the status bar has a segment that gives the user information about the status of the intellisense thread. For the stations whose subsets are neither IEEE716.89 nor CASS, this intellisense segment of the status bar shows the following message:

*Intellisense Feature Is Not Implemented For This Subset And Station.*

The meaning of this message is that the intellisense module of our Paws Studio does not have available at this time the atlas syntax definition of the current subset. In this case the intellisense module is disabled, and none of the features related to the intellisense are functional. The Paws Studio will continue to compile the atlas files as it used to, though.

Along with the atlas syntax definition available, the intellisense module needs the lexical database definition for the current station. Even for the stations part of the IEEE716.89 and CASS subsets, the default intellisense message is:

Intellisense: Your LexDB.log Is Missing. Please Rebuild Your LexDB.

The LexDB.log contains the nouns/modifiers/dimensions/descriptors information the intellisense module needs in addition to the atlas language syntax definition. If the user does not build his LexDB, the intellisense module will stay inactive. Despite this, Paws Studio and RTS will continue to work as expected. The LexDB.log is read by the intellisense module only. This is not needed by any of the standard atlas compilers, allocators or our RTS. This architecture allows the user to disable the intellisense functionality of his station by simply deleting this file.

1.3.2   New ATLAS Procedures Toolbar
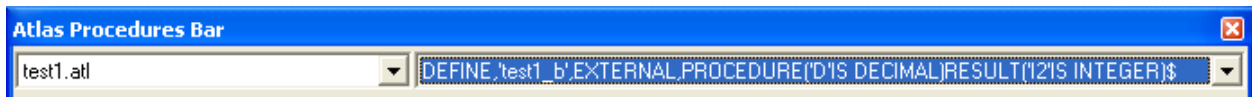
This feature is functional if the Intellisense module of Paws Studio is functional.

This new toolbar:

 1) informs the user in real time what procedure definition his caret is anytime he edits a project included atlas file. This feature is automatically triggered by the movement up and down (change of lines) of the caret.

2) allows the user to go to a specific atlas procedure definition anywhere in his project. In this case the user has to drop-down the present combo-box controls.

The Atlas Procedure Toolbar feature is complementary to and relies entirely on the good functionality of the Intellisense module explained in 1.3.1.

1.4   Problem Reports

1.4.1   PR09086:Intellisense Module Crashes When It Parses Some CASS Atlas Statements

The intellisense module of Paws Studio crashed when it parsed atlas statements that contained noun modifiers specified through the atlas syntax definition BNF rules which were not matched in the lexical database. This problem has been resolved.

2   Run Time System

Version 1.37.12
Release date: September 3, 2009

2.1   Critical Items

2.2   Known Limitations

2.3   Enhancements

2.4   Problem Reports