

**Release Notes  
ATLAS Compiler  
Version 3.23.16 (20060223)  
23 February 2006**

## 1 Overview

The following describes an overview of changes included within version 20060223 (3.23.16) of the following ATLAS compiler(s):

CASS / RT  
ESTS / PAWS  
IFTE / PAWS

and version 20060223 (3.23.16) of the following ATLAS support tools:

Signal Resource Allocator  
Switch / Ita Compiler

### 1.1 Enhancements

#### 1.1.1 Increased Paths in CNX Field

#### 1.1.2 Additional Information in Ita Compiler Path Report

### 1.2 Problem Reports

06-006, 06-007, 06-011, 06-012, 06-013, and 06-016

## 2.0 Detailed Description

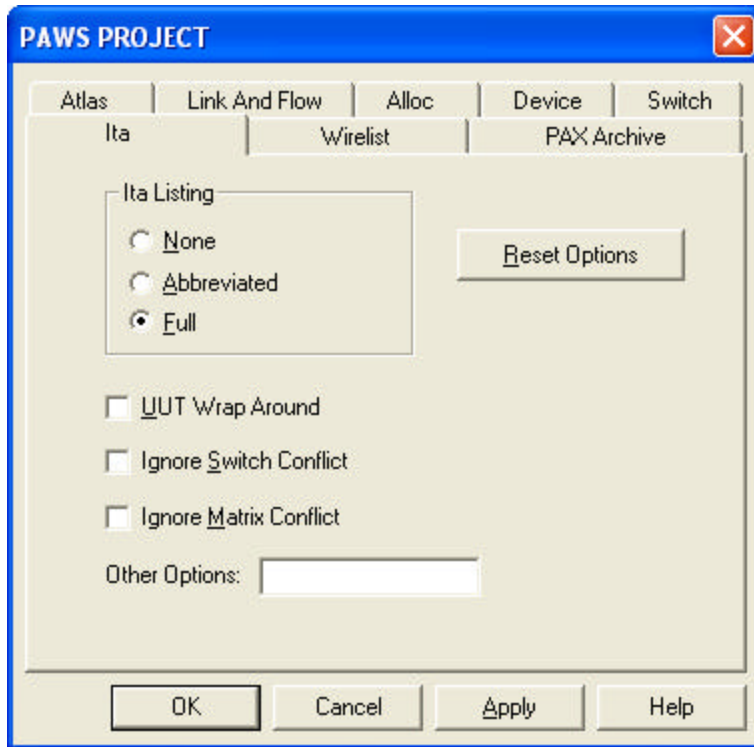
### 2.1 Enhancements

#### 2.1.1 Increased Paths in CNX Field

Previous releases of the Signal Resource Allocator limited the number of path triplets for a single CNX field to 300. This release increases that limit to 1000.

#### 2.1.2 Additional Information in Ita Compiler Path Report

The Ita compiler builds a From / To table for use by the Signal resource Allocator. Each entry in the From / To table consist of the from point, the to point and one or more sets of arc lists. Each arc list represents the switched path triplets necessary to connect the from point to the to point. A symbolic representation of the From / To table is available by using the 'Ita Listing Full' option for the build of the Ita, as shown below:



Example:

```

UUT / UUT    --- 4
UUT / PORT  --- 0
Wired       --- 0

( 1) J1-1:J2-1
      [ 1, 2]
          ( 1) BLK    1 MOD    1 PTH    1 (Switch  1:1:0)
          ( 2) BLK    1 MOD    2 PTH    1 (Switch  1:1:0)

( 2) J1-1:J2-2
      [ 1, 2]
          ( 1) BLK    1 MOD    1 PTH    1 (Switch  1:1:0)
          ( 2) BLK    1 MOD    2 PTH    2 (Switch  1:2:0)

( 3) J1-2:J2-1
      [ 1, 2]
          ( 1) BLK    1 MOD    1 PTH    2 (Switch  1:2:0)
          ( 2) BLK    1 MOD    2 PTH    1 (Switch  1:1:0)

( 4) J1-2:J2-2
      [ 1, 2]
          ( 1) BLK    1 MOD    1 PTH    2 (Switch  1:2:0)
          ( 2) BLK    1 MOD    2 PTH    2 (Switch  1:2:0)

```

Interpretation:

There are a total of 4 paths, the first of which is shown by the line:

```
( 1) J1-1:J2-1
```

The number in parenthesis is the path number. The *end points*, in this case UUT Pins are shown separated by the colon character.

The first, and in this case only, path between those *end points* is indicated by the line:

```
[ 1, 2]
```

The first number in the square brackets is the route number between the *end points*. The second numbers indicates that there are two switched *arcs* required for this route.

The lines that follow describe those *arcs*:

```
( 1) BLK      1 MOD      1 PTH      1 (Switch  1:1:0)
( 2) BLK      1 MOD      2 PTH      1 (Switch  1:1:0)
```

The number in parenthesis at the beginning of each line is the *arc* sequence number. The BLK, MOD and PTH that follow are the triplet required to activate the switch. The parenthesis at the end of the line identify the switch that is involved. The switch number, the switch ordinal position and the number of additional *gangs* for this switch.

This release of the Ita Compiler includes a new option, (Other options "f" in Ita build) that will provide, for each of the switched arcs, the end points.

### Switch / Ita Databases

```
begin BLK = 1;

  begin MOD = 1 10: PL1;
    Wire-1 to 1:10 via switch 1 PTH 1:10;
  end;

  begin MOD = 2 10: PL2;
    Wire-1 to 1:10 via switch 1 PTH 1:10;
  end;

end;

from  PL1-1  to  J1-1;
from  PL1-2  to  J1-2;

from  PL2-1  to  J1-1;
from  PL2-2  to  J1-2;
```

### Expanded From / To Table (Other Options "f")

```
Path Blocks --- 2
```

```

Total Pairs --- 1
UUT / UUT --- 1
UUT / PORT --- 0
Wired --- 0

( 1) J1-1:J1-2

[ 1, 2]
( 1) { PL2-1:Wire-1 } BLK 1 MOD 2 PTH 1 (Switch 1:1:0)
( 2) { Wire-1:PL1-2 } BLK 1 MOD 1 PTH 2 (Switch 1:2:0)
[ 2, 2]
( 1) { PL1-1:Wire-1 } BLK 1 MOD 1 PTH 1 (Switch 1:1:0)
( 2) { Wire-1:PL2-2 } BLK 1 MOD 2 PTH 2 (Switch 1:2:0)

```

The primary purpose of this enhancement is to provide the additional information required when using the ROUTE statement to select which specific route to use when connecting a pair of end points.

In the example above we can see that the first switched *arc* of the first route connects **PL2-1** to **Wire-1** and that the second switched *arc* of the first route connects **Wire-1** to **PL1-2**. To use the ROUTE statement we need also to determine how the UUT Pin **J1-1** is connected to the Interface Pin **PL1-1** and the Interface Pin **PL2-2** is connected to UUT Pin **J1-2**. This can be best achieved by examining other sections of the 'Ita Listing Full' output. In these cases, the UUT Pin section of the output is the most useful.

```

UUT Pin(s) --- 4

J1-1*=          PL1-1=          PL2-1
J1-2*=          PL1-2=          PL2-2
COMMON*        EARTH*

```

The line **J1-1\*= PL1-1= PL2-1** shows that UUT Pin **J1-1** is wired to Interface Pins **PL1-1** and **PL2-1**. Similarly the line **J1-2\*= PL1-2= PL2-2** shows that UUT Pin **J1-2** is wired to Interface Pins **PL1-2** and **PL2-2**.

Integrating the above information enables the creation of a ROUTE statement. Assuming that we wish to use the UUT Pin **J1-1** wired to Interface Pin **PL1-1** then the ROUTE statement would be as follows.

```
route ABC from J1-1 to PL1-1 to Wire-1 to PL2-2 to J1-2;
```

## 2.2 Problem Reports

### 2.2.1 06-006 - FATAL - Error Too many Structures (CASS)

Previous releases of the CASS / RT ATLAS compiler limited the number of *control structures* to 2047. This release of the subject compiler increases that limit to 32,767.

A *control structure* is a DEFINE 'procedure', an IF, FOR, WHILE, BEGIN ATLAS and BEGIN BLOCK.

### 2.2.2 06-007 - Additional Space in OUTPUT of DEFINEd MESSAGEs (ESTS)

Previous releases of the ESTS (and IFTE) ATLAS compilers included leading and trailing *whitespace* characters in OUTPUT statements that reference a DEFINE, MESSAGE 'label'.

This release of the subject compilers ignores that leading and trailing *whitespace*.

### 2.2.3 06-011 - DEFINE INTERFACE pins in Digital CNX Field (ESTS)

Previous releases of the ESTS (and IFTE) compilers incorrectly processed UUT Pins that started with a string that conflicted with a <Port> name.

Example:

```
DEFINE, INTERFACE, Clk_nTx/D88    $
...
DO, DIGITAL TEST USING 'DWG',
...
CNX-STIM Clk_nTx ...    $
```

The C character from the INTERFACE pin Clk\_nTx was processed as the <Port> C.

This release of the subject compilers requires that there be a least one blank character following any <Port>.

### 2.2.4 06-012 - Period Character in ATLAS Labels (ESTS)

In order to support an INCLUDE of a module name including the .IC or .ICS suffix and a PERFORM of that module that without referencing the suffix previous releases of the ESTS (and IFTE) compilers ignored a period and all following characters in ATLAS 'labels'. The effect of this was that if an ATLAS label included a period and it conflicted with another label then a "Duplicate definition of 'label'" WARNING message was issued.

This release of the subject compilers ignores the period and following characters only in INCLUDE / PERFORM module 'label's.

### 2.2.5 06-013 - CHR Function in Conditional Expression (ESTS)

Previous releases of the ESTS (and IFTE) compilers did not recognise a CHR() function reference in a <Conditional Expression>, e.g. FOR statement, resulting in the issuance of an ERROR message.

This release of the subject compilers recognises the use of the CHR() function in a <ConditionalExpression>.

## 2.2.6 06-016 - Digital CNX Variables in MODULES (ESTS)

Previous of the PAWS Signal Resource Allocator generated incorrect data addresses when a Digital CNX clause in an ATLAS module included CONN variables.

This release of the PAWS Signal Resource Allocator generates the correct data addresses.