

**Release Notes
ATLAS Compiler
Version 03.23.01 (20050429)
29 April 2005**

Overview

The following describes an overview of changes included within version 20050429 (3.23.1) of the following ATLAS compiler(s):

CASS / RT
CRATE / CAT, CRATE / RADCOM
IFTE / PAWS
IEEE716.89 / AllStations

and version 20050429 (3.23.1) of the following ATLAS support tools:

Signal Resource Allocator

and version 20050509 (1.5) of the CRATE Graphics Viewer.

1.1 Enhancements

CRATE Graphics Viewer Text Properties

1.2 Problem Reports

04-034
05-007
05-008
05-009
05-013
05-999

Detailed Description

2.1 Enhancements

2.1.1 CRATE Graphics Viewer Text Properties

In order to allow user configuration of text appearance, properties have been added to the Graphics Viewer to control font, font size, font **bold** or plain, independent control of width and height of font for standard and double sizes. Control in both the X and Y axes of the position of the first character within the bounding rectangle. The following describes in detail those properties.

- -DDELTA.TXT=<dX>

Controls the position of the leftmost character of the string within the bounding rectangle. <dX> is an integer value, optionally signed, that specifies the position in user pixel units. Positive values will move the text to the right, negative to the left. The default value is 0.

- -DDELTA.Y.TXT=<dY>

Controls the position of the top of the character string within the bounding rectangle. <dY> is an integer value, optionally signed, that specifies the position in user pixel units. Positive values will move the text *up*, negative *down*. The default value is 2 which causes the *top* of the character string to be flush against the top of the bounding rectangle.

- -DFONT="<FontName>"

Any legitimate *physical* Microsoft True Type font that is installed on the host computer or one of the Java logical font names, i.e. *Monospaced*, *Serif*, *SansSerif*, *Dialog* and *DialogInput*. Java *logical* fonts are mapped to *physical* fonts in a platform dependant manner. Windows systems usually use *Time New Roman* for *Serif*, *Arial* for *SansSerif* and *Dialog*, and *Courier* for *Monospaced*. The quotations are included in the property definition to allow for those font names that include space characters, e.g. *Lucida Console*. The default is *Monospaced*, though it has been reported that *Lucida Sans Unicode* closely matches that used by the legacy terminal.

- -DFONTBOLD

If set will use the bold attribute for the selected font.

- -DFONTSIZE=<fSize>

The base font point size expressed as an unsigned integer. The base font is scaled by both empirically derived internal scale factors and user properties (See `TXT{X|Y}{1|2}SCALE`). The default value is 11.

- -DTXTX1SCALE=<ScaleFactorPercent>

Apply the scale factor to the X axis of STANDARD width characters. <ScaleFactorPercent> is an unsigned integer value representing the scale factor as a percentage, the default value is 100.

- -DTXTY1SCALE=<ScaleFactorPercent>

Apply the scale factor to the Y axis of STANDARD height characters. <ScaleFactorPercent> is an unsigned integer value representing the scale factor as a percentage, the default value is 100.

- -DTXTX2SCALE=<ScaleFactorPercent>

Apply the scale factor to the X axis of DOUBLE width characters. <ScaleFactorPercent> is an unsigned integer value representing the scale factor as a percentage, the default value is 100.

· -DTXTY2SCALE=<ScaleFactorPercent>

Apply the scale factor to the Y axis of DOUBLE height characters. <ScaleFactorPercent> is an unsigned integer value representing the scale factor as a percentage, the default value is 100.

Experimentation at TYX shows that the following text properties provide visually satisfactory results, though they have not been compared with the legacy terminal.

```
-DFONT="Lucida Sans Unicode"  
-DFONTBOLD  
-DFONTSIZE=10  
-DTXTX1SCALE=75  
-DTXTX2SCALE=80
```

2.2 Problem Reports

2.2.1 04-034 IFTE / PAWS

Earlier versions of the IFTE / PAWS compiler, when processing unformatted output to a file selected a default format that allowed those files to later be read using an unformatted input statement. That default format differed from unformatted output to the station display.

This version of the IFTE / PAWS compiler processes unformatted output to both a file and the station display identically. Those files can no longer be read using an unformatted input statement.

2.2.2 05-007 IFTE / PAWS

The IFTE / PAWS compiler when encountering constant expressions evaluates them at compile time. Earlier versions of the subject compiler did not directly detect an attempt to divide by zero causing the underlying parsing machine (PLI) to issue the message: "PLI: <1056> ARITH: Divide by zero (69,113)" and compilation was terminated.

This version of the subject compiler issues the message: "Divide by zero" and compilation continues.

2.2.3 05-008 CRATE Graphics Viewer

The CRATE Graphics Viewer performance was reported as being "too slow". A baseline test that painted 625 text elements continued processing for approximately 30 seconds after ATLAS execution had completed.

This version of the Graphics Viewer reduced that time to around 2 seconds.

2.2.4 05-009 CRATE / CAT & RADCOM

All compilers with the exception of CRATE have a compiler imposed limit of 200 UUT pins / port. In order to support CRATE digital *pin ranges* that limit was increased to 255.

Release 3.23.0 of the Signal Resource Allocator required knowledge of that limit, and used the 200 value causing the issuance of the error message: “More than 200 path triplets”.

This version of the Signal Resource Allocator uses the CRATE value of 255.

2.2.5 05-013 IEEE716.89 / AllStations

Introduction of the compiler ‘y’ option, required to support the Switch Server, caused the Signal Resource Allocator to disable all *direct* paths, this resulted in the <port> to <port> paths used by the *event* architecture to be disabled.

This release of the Signal resource Allocator no longer disables those <port> to <port> paths.

2.2.6 05-999 CRATE / CAT & RADCOM

Previous versions of the compiler unconditionally set the value of the ‘TIP’ variable to that returned by the instrument driver for DO, DIGITAL TEST, STIM-ONLY statements. At customer request this version of the compiler will only set the ‘TIP’ variable when the value returned by the instrument driver is 3, i.e *timeout*.