



APL+Win Version 17.1.01
Copyright (c) 2017 APLNow LLC.
All Rights Reserved
Oct 2, 2017

The APL+Win v17.1.01 release is available and recommended for all current APL+Win subscribers. To obtain this release, visit <http://www.apl2000.com/software.php> and click [APL+Win v17.1.01 Installer](#).

Performance Improvement using SSE2 Processor Architecture

What is SSE2 Architecture

[SSE2](#) is a computer processor architecture and associated instruction set which provides for a single instruction to process multiple data items. Its application to APL+Win array operations has beneficial effects on performance.

APL+Win and SSE2 Architecture

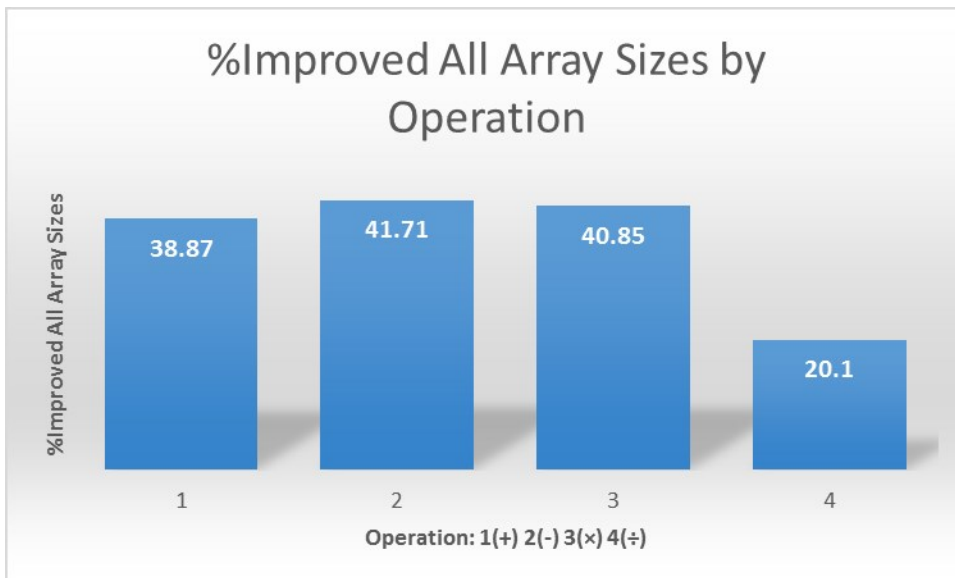
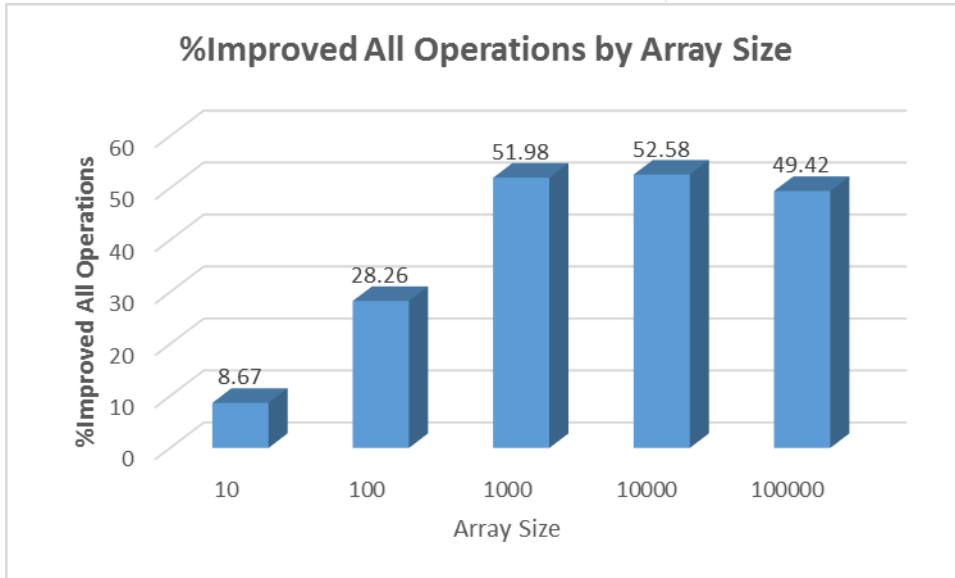
APL+Win 17.1 includes a new programmer-controlled APLW.INI configuration file setting, [Experimental] EnableSSE2. When this setting is set to 0 (the default value), the previous behavior of APL+Win will apply. When this setting is set to 1, when appropriate, the APL+Win interpreter will use the SSE2 instruction set to optimize array-based arithmetic operations, $+ - \times \div$. The DEMO_SSE2.W3 workspace is included in this release to test this enhancement.

Performance Benefits of SSE2 Architecture in APL+Win

Performance benefits of SSE2 in APL+Win depend on the operations being performed, the operand data types, the number of data elements in the operation and the workstation environment. Experimentation using an actual APL+Win application system is recommended to determine the magnitude of performance improvements with SSE2 in APL+Win. Here is a typical result set:

Corporate Office
PO Box 361
Brielle NJ 08730
732-223-5575

Sales Office
One Research Ct., Ste. 325
Rockville MD 20850
301-208-7150



Operands	%Improved All Array Sizes by Operand Data Type			
	+	-	×	÷
Float,float	34.72	41.46	26.56	42.65
fScal,float	47.44	55.53	37.28	44.99
Float,fScal	46.06	56.34	39.08	45.73

Corporate Office
PO Box 361
Brielle NJ 08730
732-223-5575

Sales Office
One Research Ct., Ste. 325
Rockville MD 20850
301-208-7150



Float,int	32.18	40.85	42.48	19.02
fScal,int	30.65	39.62	46.58	24.10
Float,iScal	46.68	35.26	52.96	15.84
Int,float	37.06	27.07	36.83	13.72
iScal,float	42.04	37.15	49.25	24.57
Int,fScal	32.96	42.07	36.62	22.30

Overall %Improved All Operations & All Array Sizes: 33.93%

Notes:

- float, int indicate floating point array and integer array respectively
- Array sizes were varied in the range [10, 100000]
- fScal, iScal indicate floating point scalar and integer scalar respectively
- Workstation: Win10 Pro, dual core 2.30Mhz

SSE2 Architecture Precision

SSE2 floating point arithmetic operations, $+ - \times \div$, can yield results that are very slightly different than traditional APL+Win result values. In most cases results will be identical. But in some cases very slight rounding differences can cause the low-order bits of the result value to be different. These possible result differences are limited to the last digit of precision in APL+Win and may not be apparent unless CT is set sufficiently small or PP is set sufficient high. However, you should be aware that result values may not be exactly the same with and without using the SSE2 optimization, even though you are unlikely to notice this difference with default CT and PP settings.

Customer Feedback is important

Since the SSE2 enhancement is currently defined as 'Experimental' in this initial release, it is intended for test purposes only. It is suggested that the SSE2 enhancement not be used in any production application until sufficient customer experience has been gained and appropriate feedback has been received and reviewed by APL2000. Please submit feedback on this enhancement to support@apl2000.com.

Bug Fixes

- Fixed bug where errors thrown by callback handlers were bypassing the :Catch control structure handler and directly exiting the function (possibly diverted through a :Finally block).

Corporate Office
PO Box 361
Brielle NJ 08730
732-223-5575

Sales Office
One Research Ct., Ste. 325
Rockville MD 20850
301-208-7150



Note: In the event this change causes any problems or unexpected results in your application, you can simply restore the original behavior by adding the setting below to your APLW.INI file:

```
[Compatibility]
CatchCallbackErrors=0
```

- In versions 15.1 through 17.0, there were two instances where the `□DEF` and `□FX` system functions incorrectly signaled a NONCE ERROR for an integer scalar argument, instead of returning (2 `□IO`) and `□IO`, respectively. For example:

```
□DEF 10
NONCE ERROR
□DEF 10
^
```

- In prior versions, it was determined that the `:Try *` statement was operating inefficiently in the interpreter that likely resulted in several hundred wasted machine instructions. This extra processing has been eliminated.
- In prior versions, the argument to the `:Try` control structure statement was not checked for validity. For example, the 10 in the `:Try` block below

```
:Try 10
  Foo
:Catch
  CatchHandler
:EndTry
```

wasn't flagged as invalid at execution time, resulting in problems when an error occurred in a function called in the `:Try` block. This bug was addressed by immediately signaling an error when the argument isn't a character vector or scalar.

APL2000
www.apl2000.com

Corporate Office
PO Box 361
Brielle NJ 08730
732-223-5575

Sales Office
One Research Ct., Ste. 325
Rockville MD 20850
301-208-7150