MS-DMT/MARS-ALE
x86/x64 Road Map

6 February 2018

Overview

This is a living document subject to change as required without prior notice. This document will be updated as required to remain current with actual software development achievements and changing development goals. This document and any update to it shall reside at:

www.n2ckh.com/MARS_ALE_FORUM/MS-DMT_MARS-ALE_x86_x64_Road_Map.pdf

The purpose of this document is to provide a high level view of the software development road map regarding both MS-DMT and MARS-ALE as 32-bit (x86) and 64-bit (x64) native executables (.EXE) and the targeted MS-Windows operating system versions and computer hardware on which their development is targeted.

The MS-DMT and MARS-ALE tools are flexible software defined alternatives to expensive hardware solutions. To attain the performance capable of being provided by these software tools, especially as related to the MIL-STD serial tone modems, it is critical that the host MS-Windows computer system on which they are used meet specified host Windows computer requirements for their use.

The current MS-DMT v2.00 x86 baseline is currently the focus of an x64 port to gain added performance benefits over that of running x86 MS-DMT v2.00 on 64-bit Windows. The MARS-ALE v3.00 tool shall remain an x86 application only. The developing MARS-ALE v4.00 tool shall debut as an x64 only application. Migrating 32-bit C/C++ code to 64-bit is not a trivial task. There are many issues to consider. However it has been determined that the x86 versions of the tools provide the best results only when running on 32-bit Windows. Thus porting the tools to native 64-bit Windows is required to achieve the best results in using 64-bit Windows for our communications needs.

It should be obvious to most that 64-bit Windows and native x64 applications provide more capability than do 32-bit Windows/x86 based applications or x86 based applications running on 64-bit Windows. However it has also been determined that x86 versions of the tools suffer in performance when running on 64-bit Windows. Herein both the challenges and benefits of 64-bit Windows with respect to our focus shall be discussed.

Windows OS Polls

There are two open ended polls on the Yahoo MARS-ALE group that enter into the planning detailed herein. Although there are over 1,600 members of the Yahoo group, less than 10% have bothered to respond to these polls to date. However I was told long ago that percentage breakdown of such polls would likely change very little if we had 100% participation in the polls, which I would like to have by the way.

Since 4 January 2016 one poll has Windows 7 users at 28%, Windows 8/8.1 at 2% and Windows 10 at 61% among respondents.

Since 5 November 2017 the other poll has 64-bit Windows users at 89% and 32-bit users at 11% among respondents.

These polls indicate that Windows 10 is now the predominant OS, with the majority being 64-bit Windows 10 users.

In addition, although not focused on in the polls, its public knowledge that over the last 10 years most computers sold were equipped with x64 processors, which can all be upgraded to 64-bit Windows 10.

As x86 applications suffer somewhat in performance on 64-bit Windows and most users are now running 64-bit Windows and Windows 10 at that, the time has come to focus on x64 development. Then too at a certain point in time to take full advantage of Windows 10 features for x64 builds to enhance their performance.

However this does not mean an end to x86 development, x86 development shall continue for a number of years yet. However it does means that it is time to take full advantage of Windows 7 features for x86 builds to enhance the tools performance.

Native x64 Benefits

The 64-bit processor was introduced in 2003 and first 64-bit Windows in 2005 and both 32-bit and 64-bit versions of Windows have been offered ever since.

Even though for over 10 years now most computers sold came with x64 processors, as 32-bit Windows has been less expensive than 64-bit Windows many end up with 32-bit Windows running on an x-64 processor.

Due to the larger real and virtual addressable memory benefits of x64 processors and over that of x86 and other benefits in general and those specific to our needs that listed herein, upgrading to 64-bit Windows is highly recommended. However at least doubling the amount of RAM when upgrading from 32-bit Windows on a 4GB 64-bit processor PC to 64-bit Windows is also highly recommended.

Any x86 application can be used on 64-bit Windows, however there are shortcomings in performance and availability of some resources in doing so. As an emulation layer is require on 64-bit Windows to run x86 applications there is some performance degradation and other issues due to added overhead. The 32-bit code running in compatibility mode may in fact run slower than a native 64-bit code. On an average 32-bit performance loss due to the emulation layer can be as little as 2-3%, although in some cases it might be much more. However performance gains going from 32-bit to a 64-bit version of an applications can be large. It depends greatly upon the type of application and its use of graphics, file I/0 and the data types they are processing. But in general you may expect a 2-20% performance gain from porting to an x64 native application.

WOW64

In particular to our interests is that running x86 MS-DMT v2.00 on 64 bit Windows many users experience start up crash issues during the Windows-On-Windows (WOW64) emulator startup stage as it starts the execution of an x86 application. The native x64 MS-DMT has no such start up issue or degraded performance as WOW64 is overhead which is eliminated by developing a native 64-bit executable.

NOTE: The Intel x64 Itanium processor architecture dating back to 2001 is the worst for running x86 applications under WOW64 as more software is involved in the emulation, and performance suffers as a result. Also, WOW64 adds significant overhead if two or more instances of the same 32-bit application are running concurrently. This is due to the native 8 KB pages on the Intel Itanium, which complicates the emulation of the native 4 KB pages on the x86 architecture (more pages are marked as writable; all writable pages are private to the process).

For x86 user-mode applications, the WOW64 thunking layer that enables 32-bit applications to execute (with some x86 application performance degradation) on 64-bit versions of Windows does so by intercepting 32-bit function calls and converting pointer-precision parameter types to fixed-precision types as appropriate before making the transition to the 64-bit kernel. Thunking is only done for 32-bit applications; 32-bit device drivers are not supported on 64-bit versions of Windows.

Thunking is done in user mode to reduce the impact on the 64-bit kernel and to reduce the risk of a bug in the thunk that might cause a kernel-mode crash, data corruption, or a security hole. The thunks extract arguments from the 32-bit stack, extend them to 64 bits, then make the native system call. Despite its outwardly similar appearance on all versions of 64-bit Windows, WOW64's implementation varies depending on the target processor architecture. The newer the processor the better the WOW64 performance.

MEMORY

A 64-bit processor supports considerably larger amounts of virtual memory and physical memory than is possible on 32-bit architectures. This is one of the major benefits of 64-bit, because it allows programs to store large amounts of data in memory. In addition, x86-64 provides 64-bit general-purpose registers and a variety of other enhancements, such as a more resilient memory model.

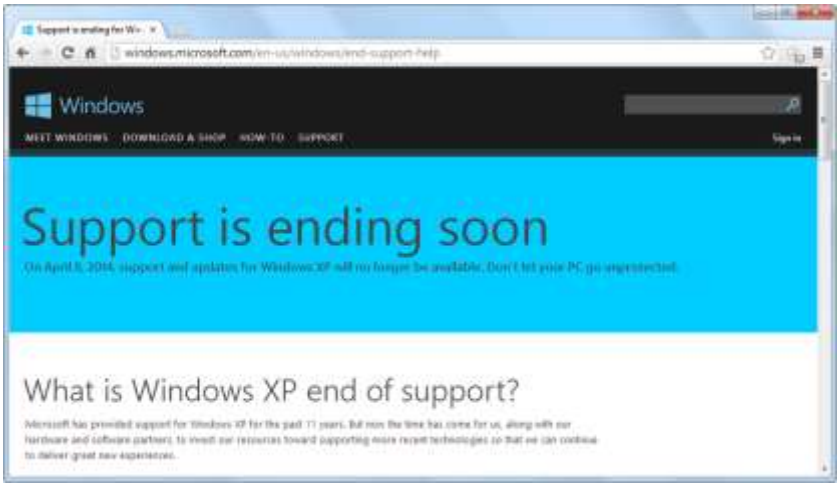
An x64 native application is memory efficient, whereas an x86 application running on 64-bit Windows is not so much. An x86 process that accesses a disk, the network, the registry, or any other system resource under WOW64 results in the corresponding API call invariably ending in kernel mode. The x64 kernel, of course, expects 64-bit data structures (pointers and the like). 32-bit applications, on the other hand, use 32-bit data structures which requires conversion (called thunking), which is performed by the WOW DLLs mentioned above, but it comes at a price. Not so much in performance as CPU overhead is mostly negligible in most cases, but at the cost of required RAM. Converting the 32-bit data to 64-bits essentially doubles the amount of RAM required to store the data.

However, since only some data structures need to be converted and use of those structures varies greatly between applications, it is simply not possible to give a good estimate of how much more RAM is needed to run a 32-bit app on x64 than on 32-bit Windows. The memory overhead increases with the complexity of an applications (more complex = more overhead). Typical multipliers for the memory consumption of single processes range from 1.2 – 2.5 based on the memory consumption seen on a 32bit Windows OS. Thus a 64-bit PC with 4GB RAM running 32-bit Windows that is upgraded to 64-bit Windows where x86 applications continue in use may actually suffer somewhat running under WOW64 due to too little RAM.

MULTI-THREADING

As MS-DMT and MARS-ALE are multi-threaded applications, another benefit is that x86 applications may not be able to create as many threads under WOW64 as they can when running natively on 32-bit systems. This is because WOW64 allocates an additional 64-bit stack (usually 512 KB) for each thread. In addition, some amount of address space is reserved for WOW64 itself and the data structures it uses. However there is no such limitation with an x64 native application.

Supported Windows OS Hosts



The Windows 7 operating system versions will soon become the minimum versions of Windows supported by both x86 and x64 executables. This will come about when both MS-DMT and MARS-ALE are enhanced by taking advantage of specific Windows 7 features that have not previously been used so as to permit the tools to work on older legacy Windows versions.

The specific targeting of Windows 7 features that will provide enhanced tool performance and force the use of Windows 7 and hardware on which it will run is the next step in the evolution of the x86 development and for the new developing x64 versions of the tools.

There are no plans to take advantage of features beyond those of Windows 7 in the x86 tools. This means that the x86 tools will only become obsolete when required operational features are no longer being developed for the x86 based tools.

At least Windows 7 and computer hardware designed for running Windows 7 or at a minimum for running Windows Vista is required for MS-DMT and MARS-ALE at this point. For later versions of Windows it is also true that PC hardware developed for the current version of Windows yields the best results with the next best results being had from one generation older PC processor hardware and supporting chip sets.

NOTE: All testing for performance is performed on under an MS-Windows OS as described below on "Air-Gap" computers with no anti-virus or other such software running.

NOTE: No Spectre/Meltdown mitigation patches or BIOS updates for the same are installed on any computer used for performance testing.

NOTE: It is recommended that all use of the x86 and x64 tools be made on "Air-Gap" computers unless Windows 10 is being used with a processor (CPU) made in 2016 or later, in which case Spectre/Meltdown mitigation software patches and BIOS update patches can be installed as such patches result in minimum performance degradation on such systems.

The following breaks down supported and recommended Windows OS's by x86 and x64 builds and planned development:

x86 tools-

Both 32-bit and 64-bit Windows 7 versions are the minimum Windows OS that the x86 MS-DMT v2.00 and x86 MARS-ALE v3.00 tools are being designed to use. However Windows 10 versions are recommended for use of the tools.

Developer based testing of the x86 tools is currently performed on 32-bit Windows 7, Windows 8, Windows 8.1 and 64-bit Windows 10 operating systems.

The active development of x86 tools will likely not come to an end until at least 2021, a full 3 years from now. The x86 tools will continue receive bug fixes and minor new feature development additions for some period of time at the conclusion of active development until it is decided to make a final release.

x64 tools -

64-bit Windows 10 versions are the recommended Windows OS for x64 MS-DMT v2.00.

All developer based testing is currently performed on Windows 10 Home and Professional only. Thus no guarantee can be made of support on other than Windows 10 versions. However the tool should work on back to 64-bit Windows 7 versions at this time.

The x64 applications will start out by taking advantage of specific Windows 7 features for enhanced performance. However by the time x64 MARS-ALE v4.00 is released the intent is to be taking advantage of specific Windows 10 features for both x64 MS-DMT and x64 MARS-ALE for enhanced performance. This will then position Windows 10 as the minimum 64-bit Windows OS.

At present the estimate is that x64 MARS-ALE v4.0 will debut for initial testing by the end of 2020 with a full release some time during 2021.

Supported CPU's

All 32-bit and 64-bit processors made since the year 2000 that are Streaming SIMD Extensions 2 (SSE2) capable, which can run Windows 7 or later are currently supported.

The use of computer hardware greater than 10 years of age can be tolerated as long we are talking use of hardware no older than one generation Windows prior, e.g. hardware shipped for Windows Vista being used with a clean install of Windows 7, hardware shipped with Windows 7 being used with Windows 8, etc.

x86 tools -

The x86 builds of the tools shall continue to only require hardware support for SSE2.

x64 tools -

At some point during the x64 development cycle support of Advanced Vector Extensions (AVX) may come about. AVX is supported from Windows 7 SP1 onward. The support of AVX in the tools may coincide with the debut of MARS-ALE v4.0, tentatively in 2021.

AVX support will require use of processor's made since 2011 which support AVX, which by then means the use of computers that can be as much as 10+ years old.

Current list of AVX supported processor architectures includes released/coming CPU families:

Intel -

- Sandy Bridge processor, 2011
- Sandy Bridge E processor, 2011
- Ivy Bridge processor, 2012
- Ivy Bridge E processor, 2013
- Haswell processor, 2013
- Haswell E processor, 2014
- Broadwell processor, 2014
- Broadwell E processor, 2016
- Skylake processor, 2015
- Kaby Lake processor ULV mobile, 2016
- Kaby Lake processor desktop/mobile, 2017
- Coffee Lake processor, 2017
- Cannonlake processor, expected in 2017
- Cascade Lake processor, expected in 2018
- Ice Lake processor, expected in 2018
- TBD processors…

AMD -

- Bulldozer-based processor, 2011
- Piledriver-based processor, 2012
- Jaguar-based processor. 2013
- Steamroller-based processor, 2014
- Puma-based processor, 2014
- Excavator-based processor, 2015
- Zen-based processor, 2017
- TBD processors…