■■■ white paper

Rocket®

The 64-Bit Question: What Does It Mean for Rocket U2 Customers?

The 64-Bit Question: What Does It Mean for Rocket U2 Customers?

August 2010
Version 5.0
Revised September 2014

Note: All the information contained in this document is based on publicly available information and is subject to change. Rocket Software disclaims all warranties as to the accuracy, completeness, or adequacy of such information. Rocket Software shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof.

The 64-Bit Question: What Does It Mean for Rocket U2 Customers?

What are the advantages of using 64-bit files, memory addresses, data types, or 64-bit compiled versions of Rocket UniData or Rocket UniVerse (U2)? What, if anything, is gained from moving from 32-bit to 64-bit in any of these areas? This document reviews the specific advantages, if they exist, in each of these areas and explains how each relates to the Rocket Software's nested relational database products.

64-Bit Database Files

Within our database hashed file structure, pointers (offset from first byte in the file) exist to link blocks of data together within the file. These pointers can either be 64-bit or 32-bit signed integers and each has a limit to how big a single file can grow. A 64-bit database file allows a hashed file to be up to 9 million terabytes in size and thus greatly overcomes the 2GB 32-bit single file limit. With demands on modern applications to store ever increasing data volumes, the limit of 2GB on 32-bit static-hashed database file storage can be easily reached. With the vastly increased speed of modern processors and storages systems, the small advantage of the more compact 32-bit pointers is negligible, and the risk of overflowing the 32-bit structure is substantial. When this limit is reached, there can be no automatic conversion to 64-bit file-type and the application update will fail, resulting in application failure and lost productivity. With the UniData 32-bit dynamically-hashed database file, the 2GB limit could be exceeded, but the overhead of opening/updating multiple partition files was substantial.

UniData 8.1 introduces 64-bit database files on all of its supported platforms. This support is for both static-hashed and dynamically-hashed files which removes the worry of hitting the 2GB limit, and the overhead of opening multiple files.

64-Bit Memory Addressing / 64-Bit Compilation

When a program is compiled it can be targeted for either a 32-bit address-space or a 64-bit address space. In the past, processor architecture was limited to addressing only 32-bit memory (2GB to 4GB depending on OS implementation). All modern CPUs now support 64-bit addressing, but each platform or vendor released 64-bit support on a different timeline. We targeted (or built) UniVerse and UniData for 32-bit addressing on some platforms and 64-bit on others depending on how advanced the platforms architecture was at the time.

A 32-bit targeted program can run on a 64-bit processor/operating system by using a "compatibility-mode", however the converse is not possible. For this reason, all of the current 32-bit U2 versions support running on 64-bit versions of Linux, UNIX, and Windows in compatibility mode.

Starting at UniVerse 11.2.3, 64-bit memory addressing was added for Windows, Linux, Solaris Sparc, and Solaris x86. We continue to build 32-bit versions for these platforms for 11.2 life-cycle, but will drop all 32-bit targets for the UniVerse 12 release.

Starting at UniData 8.1 release all builds will be exclusively 64-bit targeted. This means that older Windows, Linux, and Solaris installations that were installed with 32-bit O/S versions (even if the processor was 64-bit) will not be supported.

When U2 is running a 32-bit targeted build, each process is limited to addressing 2GB for its program variables. With many processes (users) running on a 64-bit O/S, more than 2GB could be in use at any one time; However if an individual U2 BASIC program (process) tries to load a large number of records into its memory for processing that exceeds 2GB, a limit would be reached in which the program would terminate. environment. Currently this capability is not supported in the U2 engines. However, many customers have successfully used U2 BASIC's string mathematics capability to achieve the same functionality.

This functionality is being considered for a future release of each data server product.

Summary

UniVerse and UniData have come from a past where 32-bit was the predominate architecture for both files and memory addressing. With the latest releases of the products, they have moved to a fully 64-bit architectures in every major aspect. With a 64-bit architecture, virtually all practical limits have been removed. To put it simply, this change represents an increase of 4,000,000,000 times the size of the previous limits.

white paper ■ ■ ■

Rocket®

🌐 www.rocketsoftware.com

✉ info@rocketsoftware.com