

Rocket MultiValue Platform and Virtual Machine Support

Rocket MultiValue Platform Support

The MultiValue (MV) application servers (i.e., UniVerse, UniData, D3, jBASE, OpenQM) are available on a wide variety of hardware platforms (x86, Power, etc.), processor cores, and operating systems (O/S) such as Linux, AIX and Windows. We make decisions about which hardware platforms and O/Ss to support on an on-going basis; we consider market demands and O/S lifecycles.

We also consider market requirements in conjunction with our binary compatibility statement when prioritizing porting projects. MV features that we've built into a specific version of the application server follow the same O/S platform support matrix as the application server.

For information on O/Ss and versions supported on specific MV application server versions, check the Rocket MV Product Availability Matrix (PAM) at: <https://rbc.rocketsoftware.com/matrix.asp>

If the O/S version indicates All in RBC, see the product documentation for more specific information on the minimum operating system version we recommend.

For any questions on O/S support, send an email to Rocket Business Connect:
rbc@rocketsoftware.com

The MV platform includes tools such as MultiValue Integration Server (MVIS) and Rocket SystemBuilder Extensible Architecture (SB/XA). The MV tools can be categorized as:

- MV Tools that we built using only the MV Basic language that will run on the version of the application server to which they were ported and any higher version unless otherwise noted.
- MV Tools or components that include low level language compiled to the machine level as well as BASIC which is compiled to the pcode level that will run on the specific O/S and application server versions noted and any higher level unless it is not binary compatible or is otherwise noted.

Binary Compatibility

Our MV Binary Compatibility statement covers MV Products which we've ported at the Operating System (O/S) level. This excludes SB+ Server, which we've ported at the MV application server version level. We generally port MV to the lowest possible vendor-supported O/S version to support the broadest base of users. Due to resource limitations, we do not install each service pack and patch level of each O/S we support when we announce application server certification. Instead, we depend on the vendor's statement of binary compatibility.

If the vendor indicates that the latest release of their O/S has binary compatibility with an earlier version, we expect that our application servers will run at that O/S version. If the vendor has changed its O/S major version number (e.g., AIX 6x to 7x), or we have concerns based on past or current field experience, we will install the new O/S version and run our QA certification before initially confirming binary compatibility.

Version dependencies for linking in external C routines may limit the extent of binary compatibility. You may need to compile external C routines on the same version of the O/S on which the MV application server was built.

If, for any reason, we find that binary compatibility fails, we will attempt to work around the binary compatibility problem in our code and re-build the program on the earliest possible release level so it will support that and subsequent release levels.

Paper Certification

Rocket MV may, at our discretion, choose to paper certify an O/S. We don't test a paper certified O/S, but we support it based on the ability to reproduce issues on the fully QA certified O/S for which it has binary compatibility. To obtain support, customers must be able to reproduce any issues on a fully certified O/S.

Linking Limitations

When we build an MV product on a certain O/S version and certify that it will run on a higher version based on our binary compatibility statement, you may experience limitations linking in C routines (unless the C routines are built at the same O/S version on which the MV application server was built) via:

- CallBasic
- InterCall and CallC for UCI
- InterCall and GCI for UniVerse

The ability to link in C routines compiled at a different O/S version than the MV application server depends solely on the O/S vendor's compiler options and stated support for mixed version linking. Note: If MV is compiled as a 64-bit binary, any C routines you linked to it must also be compiled as 64-bit binaries.

O/S Version and Platform Support Statement

We support the following primary porting platforms for Rocket MV products:

- AIX on Power
- Windows on x86
- Redhat Linux on x86

We also support products on HPUX on Itanium, Solaris on Sparc and, for UniVerse, Solaris on x86. Please review the PAM (<https://rbc.rocketsoftware.com/matrix.asp>) or send an email to Rocket Business Connect (rbc@rocketsoftware.com) if you have a business need to see what MV product versions are ported to these platforms.

We may withdraw support for operating systems or O/S versions under the following circumstances:

- The O/S version is no longer available from the vendor. Please see the vendor's web site for information on their product lifecycle.
- The O/S version is no longer supported by the vendor. Please see the vendor's web site for information on their product lifecycle.
- A given O/S has been discontinued by the vendor.

- The Rocket MV team has determined there is no longer a business case to continue porting on a given O/S.
- Lack of binary compatibility between O/Ss

Windows

We generally support Rocket MV products that run on Windows at the latest supported releases of Windows (e.g., Windows Server 2019, 2022 and Windows 10/11).

For information on Microsoft support for Windows versions, see the Windows Lifecycle at <https://docs.microsoft.com/en-us/lifecycle> .

Please note that Microsoft Windows comes in several editions or packages, and we test a wide variety of combinations. While we may not test every possible permutation of version/edition/package, our certification covers most business-level editions or packages of Windows.

Linux

Current releases of MV products that we've ported on Linux include the distribution details including glibc and kernel versions.

Rocket MultiValue Support for Virtualization

This statement covers the virtualization of hardware, software and desktop environments. Examples of virtualization technologies include AIX® DLPARs, Citrix, and VMware® ESX Server.

Every instance of Rocket MV application server software in a virtualized environment must have its own unique license.

MV products leverage binary compatibility provided by virtualization technologies.

If you submit a standard usage or defect-related service request, and your software is running in a virtual environment, Rocket MV Technical Support will assume that the problem is common to both non-virtual and virtual environments. If Rocket MV Technical Support staff believes that a problem is the result of the virtualization technology or environment, we will require you to recreate the problem in a native environment before providing continued defect support.

Rocket Software can make no guarantees with respect to performance or scalability in a virtualized environment.

Setup, configuration and tuning of virtual environments are not part of standard Rocket MV Support. If you need support for setting up a virtual environment, you should consult the appropriate vendors and the Rocket Customer Solutioning team (solutioning@rocketsoftware.com). This virtualization environment support statement applies to any virtualization technology providing binary compatibility between a non-virtualized and virtualized environment.

For additional information about a specific virtual environment, contact the vendor of the virtualization technology.