BROCHURE // MULTIVALUE

Rocket® software

Upgrade Best Practices

Actionable guidance on upgrading your Rocket® MultiValue Application Platform

Planning an upgrade to your Rocket® MultiValue application platform for added features, security, performance, or to meet a particular business need?

Follow the best practices to gain actionable guidance on upgrading your Rocket MultiValue Application Platform.

01

Assess and plan thoroughly

- **Inventory customizations:** We recommend documenting all customizations, third-party integrations, and dependencies in your Rocket® MultiValue environment.

  NOTE: If your application interacts with third-party functionalities like OpenSSL, Python, or XML, and the data server upgrade includes upgrades to these third-party integrations, you'll need to thoroughly test (see below) before going live. If you've written C functions that you call from BASIC, please note that you'll need the source code and access to a compiler on the new target system.

Pro tip: Don't forget about your operating system. You can see if it's supported on the version you're upgrading to by visiting the Product Availability Matrix (PAM).

- **Review release notes and notices:** It's important to carefully analyze Rocket's Release Notes to identify new features, deprecated functionalities, bug fixes, and compatibility changes. We recommend reviewing **every** release between your current version and your upgrade target. Release Notes are specific to each version. For example, if you upgrade from UniVerse 11.2.5 to 11.3.5 but only review the 11.3.5 release notes, you might miss essential changes introduced in versions in between. Release Notes and Notices for each version are available on the PAM, so you can easily access all the information you need about your upgrade path.

- **Impact analysis:** Assess how changes may affect your application stack and custom code. Prioritize areas of potential risk.

Pro tip: If you routinely change default VOC items, remember to reapply these changes after the upgrade.

- **Consider signing up for a Discovery Assessment or purchasing a HealthCheck** to prepare for your upgrade project.

Pro tip: Work with your Rocket Sales rep to order evaluation licenses for your upgrade version.

02

Engage stakeholders early

- **Cross-functional input:** Involve application owners, database administrators (DBAs), developers, and infrastructure teams from the start.
- **Partner with Rocket:** Contact professional services for dedicated assistance with the upgrade process. Rocket Support can answer questions about the upgrade process.

Pro tip: Submit a ticket to the Rocket Tech Support team for the dates of your upgrade to inform our team.

03

Establish a robust testing strategy

- **Build a mirror test environment:** Create a replica of your production environment to test the upgrade without affecting your live systems.
- **Automate regression tests:** Where possible, use automated test scripts to validate key functionality, business processes, and interfaces.
- **Stress and performance testing:** Validate that the new version meets performance expectations under realistic loads.
- **Test boundary applications:** Validate that any integration with the operating system or third-party solutions integrated into your MultiValue applications is tested and verified by the owner of the third-party solution.

NOTE: The amount of testing depends on the ‘delta’ between releases. For example, upgrading from UniVerse 11.3.4 to 11.3.5 is relatively straightforward. However, moving from UniVerse 11.2.5 to 14.1.1 is likely to be more challenging.

Pro tip: Test, test, and test.
You should be trying to break it!

04

Backup and rollback planning

- **Comprehensive backup:** Back up databases, configuration files, and system settings before beginning the upgrade. For example, if you've changed "standard" functionality in any files, back up this configuration so you can reapply the changes after the upgrade.

  NOTE: Make a backup copy of any changed files such as uvhome as a safeguard.

- **Rollback strategy:** Document a clear rollback plan in case the upgrade needs to be reversed quickly.

05

Stage the rollout

- **Upgrade validation phase:** If possible, test the upgrade on a non-production server prior to going live.
- **Phased implementation:** If feasible, perform a phased rollout starting with non-critical environments or subsets of users.
- **Monitor and adjust:** Closely monitor logs, performance metrics, and user feedback after each phase.

06

Update documentation and training

- **Internal documentation:** Update operational guides, configuration records, and application architecture diagrams.
- **End-user and admin training:** Provide tailored training on new features or changes in system behavior.

07

Leverage Rocket resources

- **Rocket community and knowledge base:** Use Rocket's online community, webinars, and technical articles for insights and troubleshooting tips.
- **Stay current:** Subscribe to release updates to remain informed about future enhancements and security patches.
- **Inform support:** While Rocket tech support has weekend coverage for emergencies, we recommend opening a support case to let support know when you plan on upgrading. While we do not expect you to run into issues, if you follow these best practices, it's still a good idea to let Rocket know so we can be prepared as well.

Remember, your MV-based application is unique and has grown and changed with your organization's needs over the years or decades. While no two upgrades are identical, these best practices can help your upgrade project go more smoothly.

Are you ready to upgrade?

Connect with your Rocket Account Executive or visit our Upgrade Resource Page to get started.

About Rocket Software

Rocket Software is the global technology leader in modernization and partner of choice that empowers the world's leading businesses on their modernization journeys, spanning core systems to the cloud. Trusted by over 12,500 customers and 750 partners, and with more than 3,000 global employees, Rocket Software enables customers to maximize their data, applications, and infrastructure to deliver critical services that power our modern world. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located around the world. Rocket Software is a portfolio company of Bain Capital Private Equity. Follow Rocket Software on LinkedIn and X (Formerly Twitter).

Visit RocketSoftware.com ›

Rocket® software

Modernization. Without Disruption.™