jDeploy 5.5 Release Notes

Table of Contents

Overview

jDeploy 5.5 introduces launcher version enforcement, enabling applications to require minimum launcher versions and ensuring users can adopt new jDeploy features with confidence.

What’s New in 5.5

Launcher Version Enforcement

Applications can now enforce minimum launcher version requirements using the `minLauncherInitialAppVersion` property, ensuring compatibility when adopting new jDeploy features or addressing critical launcher updates.

The Problem

A jDeploy application consists of three components:

- **The Installer**: Downloads and installs the application
- **The Launcher**: Native binary that bootstraps and updates the application
- **The Java Application**: The actual JAR files

When the launcher performs automatic updates, it can update the Java application and even the JVM itself, but it cannot update itself. This limitation prevented applications from requiring newer launcher features without breaking compatibility.

The Solution

The `minLauncherInitialAppVersion` property allows you to specify the minimum application version whose launcher is required to run your current version.

Configuration

Specify the minimum launcher version in package.json:

```
{
  "name": "my-app",
  "version": "1.2.0",
  "jdeploy": {
```

```
    "minLauncherInitialAppVersion": "1.1.0"
  }
}
```

How It Works

1. **Launcher tracks its initial version**: Each launcher knows the application version it was created with
2. **Application specifies requirements**: Your package.json specifies the minimum launcher version needed
3. **Version check at launch**: The launcher compares versions before starting the application
4. **User prompt**: If the launcher is too old, users are presented with a dialog offering three choices:
   - **Update Now**: Automatically downloads and launches the latest installer
   - **Skip This Version**: Won't be prompted again for this version
   - **Ask Later**: Re-prompts in 1 week

Example Scenario

Your application release history:

- **1.0.0** - Initial release
- **1.1.0** - Added new features, launcher unchanged
- **1.2.0** - Requires new jDeploy 5.5 launcher features

For version 1.2.0, set:

```
{
  "jdeploy": {
    "minLauncherInitialAppVersion": "1.1.0"
  }
}
```

User Impact:

- Users who installed at version 1.1.0 or later: □ Can run version 1.2.0 immediately
- Users who installed at version 1.0.0: □□ Prompted to update launcher with convenient options

Update Dialog

When a launcher version check fails, users see a dialog with the following information and options:

```
This version of MyApp requires an updated launcher.

Your launcher version: 1.0.0
```

```
Required launcher version: 1.1.0

[Update Now]  [Skip This Version]  [Ask Later]
```

User Options:

- **Update Now**: Automatically downloads and launches the latest installer. The user can complete the installation with minimal effort.
- **Skip This Version**: Dismisses the dialog and launches the application anyway. The user won't be prompted again for this specific version, but will be prompted if a future version has a higher launcher requirement.
- **Ask Later**: Dismisses the dialog and launches the application. The prompt will reappear the next time the user launches the application after 1 week has passed.

Branch-Based Releases

This property is **ignored** for branch-based releases (development/testing channels) to avoid interfering with rapid iteration and testing workflows.

Use Cases

- **Adopting new jDeploy features**: Require launchers that support new capabilities
- **Security updates**: Force users to update launchers with security fixes
- **Breaking changes**: Ensure compatibility when application architecture changes
- **JVM requirement changes**: Handle launchers that need different bootstrapping logic

Best Practices

When to Use minLauncherInitialAppVersion

✅ DO use when:

- Adopting new jDeploy features that require launcher support
- Fixing critical security issues in the launcher
- Making breaking changes to application architecture
- Changing JVM requirements that affect launcher behavior

❌ DON'T use when:

- Making routine application updates
- Changing application code that doesn't affect the launcher
- Updating dependencies that don't affect bootstrapping
- Testing new features (use branch-based releases instead)

Version Selection Strategies

Conservative Approach (Recommended)

```
{
  "minLauncherInitialAppVersion": "1.1.0"  // Last known good version
}
```

Allows the widest possible user base while enforcing minimum requirements.

Aggressive Approach

```
{
  "minLauncherInitialAppVersion": "2.0.0"  // Current version
}
```

Prompts all users to update to the latest launcher (regardless of their current version), which may be inconvenient for users with working installations.

User Communication

When setting this property:

1. **Release Notes**: Clearly document the launcher requirement
2. **Migration Guide**: Provide step-by-step reinstallation instructions
3. **Download Links**: Include direct links to installers
4. **Changelog**: Explain why the launcher update is needed

Example Release Communication

```
## Version 2.0.0 - Launcher Update Recommended

**Important**: This version works best with an updated launcher. When you
launch the application, you'll be prompted to update with a convenient
one-click installer download.

### Why?

Version 2.0.0 uses new jDeploy 5.5 features that require an updated
launcher to function properly.

### How to Update

When prompted:
```

```
1. Click "Update Now" to automatically download and run the installer
2. Or download manually from: https://github.com/yourorg/myapp/releases/latest
3. Your data and settings will be preserved

You can also choose to skip this version or postpone the update if needed.

### What's New

- [Feature descriptions here]
```

Technical Details

Version Comparison

Uses standard semantic versioning (semver):

- `1.2.3 > 1.2.2`
- `1.3.0 > 1.2.9`
- `2.0.0 > 1.9.9`
- `1.2.3-beta < 1.2.3`

Pre-release versions are compared according to semver rules.

Launcher Metadata

The launcher binary contains embedded metadata:

- **Initial App Version**: Version from package.json when launcher was created (immutable)
- **Current App Version**: Updated dynamically when app updates are installed

Implementation

When the launcher starts:

1. Reads `minLauncherInitialAppVersion` from package.json
2. Compares with its embedded initial version
3. If requirement not met:
    - Displays update dialog with application name and version info
    - Provides three user options:
        - **Update Now**: Downloads and launches the installer automatically
        - **Skip This Version**: Records the skip preference and launches the application
        - **Ask Later**: Records the timestamp and launches the application (will prompt again after

1 week)

4. If requirement met:
    - Proceeds with normal application startup

User Choice Persistence

The launcher tracks user choices for update prompts:

- **Skip decisions**: Stored per-version, so users won't be prompted again for the same `minLauncherInitialAppVersion` value
- **Ask later timestamps**: Stored with 1-week expiration, after which the prompt reappears
- **Update now**: Immediately downloads and launches the installer for the latest version

Platform Support

This feature works on all platforms:

- macOS (x64 and ARM64)
- Windows (x64 and ARM64)
- Linux (x64 and ARM64)

Migration Guide

For Application Developers

Adding the Property

1. Determine the minimum launcher version needed
2. Add to package.json:

```
{
  "jdeploy": {
    "minLauncherInitialAppVersion": "X.Y.Z"
  }
}
```

3. Update release notes with upgrade instructions
4. Test with both old and new launchers

Testing

1. Install your app at the minimum required version

2. Update to the new version
3. Verify it launches successfully
4. Install your app at an older version
5. Update to the new version
6. Verify the error message appears

For End Users

When encountering a launcher version prompt, you have three options:

Option 1: Update Now (Recommended)

1. Click **Update Now** in the dialog
2. The installer will automatically download and launch
3. Follow the installer prompts to complete the update
4. Your application data and settings are preserved

This is the recommended option as it ensures you have the latest features and security updates.

Option 2: Skip This Version

1. Click **Skip This Version** if you don't want to update
2. The application will launch normally
3. You won't be prompted again for this specific version requirement

Note: Some features may not work correctly if the launcher is too old. This option is best for users who need to defer the update for specific reasons.

Option 3: Ask Later

1. Click **Ask Later** if you want to postpone the decision
2. The application will launch normally
3. You'll be prompted again in 1 week

Note: This is useful if you're in the middle of something and want to update at a more convenient time.

Upgrading from 5.4

jDeploy 5.5 is backward compatible with 5.4 and earlier versions. Applications that don't use the new `minLauncherInitialAppVersion` property will continue to work exactly as before.

To start using launcher version enforcement:

1. Upgrade to jDeploy 5.5
2. Add the `minLauncherInitialAppVersion` property when needed
3. Document the requirement in release notes
4. Test the upgrade flow with users

Advanced Usage

Strongly Encouraging Updates

To prompt all users to update to the latest launcher:

```
{
  "name": "my-app",
  "version": "2.0.0",
  "jdeploy": {
    "minLauncherInitialAppVersion": "2.0.0"
  }
}
```

This prompts all users (regardless of their current launcher version) to update to the launcher that ships with version 2.0.0, providing them with the "Update Now" option for a convenient one-click update.

Gradual Migration

For gradual adoption of new features:

- **Version 1.5.0**: Add new optional features (no launcher requirement)
- **Version 1.6.0**: Make features default but still optional
- **Version 2.0.0**: Require features with `minLauncherInitialAppVersion: "1.5.0"`

This gives users time to naturally upgrade before enforcement.

Testing Launcher Requirements

To test launcher version enforcement without affecting production:

1. Use a test application or branch-based release
2. Set `minLauncherInitialAppVersion` to a future version
3. Verify the error message displays correctly
4. Test the reinstallation workflow
5. Adjust messaging as needed

Security Considerations

Forcing Security Updates

If a security vulnerability is discovered in launchers before a certain version:

```
{
  "jdeploy": {
    "minLauncherInitialAppVersion": "1.5.0"  // First secure version
  }
}
```

This prompts users with older launchers to update. While users can choose to skip or postpone the update, the dialog makes them aware of the security issue and provides a convenient one-click update option.

Phased Security Rollouts

1. **Phase 1**: Release patched launcher as normal update (version 1.5.0)
2. **Phase 2**: Give users time to naturally upgrade (1-2 weeks)
3. **Phase 3**: Enforce requirement in next release (version 1.5.1+)

This minimizes disruption while ensuring security.

Related Features

Existing Version Properties

- `version`: Application version (standard npm property)
- `jdeploy.javaVersion`: Minimum Java version requirement
- `jdeploy.javafx`: JavaFX version requirement
- `jdeploy.javaFxVersion`: Specific JavaFX version

Complementary Features

- **Installer Generation**: Creates installers with embedded launcher
- **Automatic Updates**: Updates app files without launcher
- **JVM Management**: Downloads and manages Java installations
- **Branch-Based Releases**: Development and testing channels

Limitations

What This Does

 Prompts users when launcher is out of date Automatically downloads and launches installer when user clicks "Update Now" Provides user choice (update now, skip version, or ask later) Allows developers to adopt new jDeploy features with confidence Works with both GitHub and npm-based updates Remembers user preferences (skip decisions and ask-later timestamps)

What This Does NOT Do

 Does not force immediate updates (users can skip or postpone) Does not block the application from launching (user can choose to skip) Does not work with branch-based releases (intentionally ignored) Does not version-check individual launcher features (only overall launcher version)

Why No Automatic Launcher Updates?

Launchers cannot self-update because:

- Native binaries cannot safely replace themselves while running
- Different platforms have different executable formats
- Code signing and security requirements vary by platform
- Atomic updates of running executables are unreliable

Manual reinstallation ensures:

- Clean, verified launcher replacement
- Proper code signing on all platforms
- Atomic installation with rollback capability
- User awareness of system-level changes

Examples

Example 1: New jDeploy Feature Adoption

Your app wants to use HTML splash screens (introduced in jDeploy 5.4):

```
{
  "name": "my-gui-app",
  "version": "3.0.0",
  "jdeploy": {
    "minLauncherInitialAppVersion": "2.9.0"  // Last version before 3.0.0
  }
```

```
}
```

Users with launchers from 2.9.0 or later can use the new splash screens.

Example 2: Security Fix

A security issue was fixed in the launcher that shipped with version 1.4.0:

```
{
  "name": "secure-app",
  "version": "1.4.1",
  "jdeploy": {
    "minLauncherInitialAppVersion": "1.4.0"  // First patched version
  }
}
```

Users with older launchers will be prompted to update to the patched version, with a convenient one-click installer download available.

Example 3: Breaking Architecture Change

Your app changes from standard JRE to JBR (JetBrains Runtime):

```
{
  "name": "my-app",
  "version": "4.0.0",
  "jdeploy": {
    "jdkProvider": "jbr",
    "jbrVariant": "jcef",
    "minLauncherInitialAppVersion": "4.0.0"  // Requires JBR-aware launcher
  }
}
```

Prompts all users to update to a launcher that understands JBR configuration. Users with older launchers will see the update dialog with a one-click installer option.

Troubleshooting

"Launcher Version Required" Error

Symptom: Application shows error about launcher version

Solution:

1. Download the latest installer from the provided URL

2. Run the installer (your data is preserved)
3. Launch the application

Version Comparison Issues

Problem: Launcher version seems newer but still fails check

Cause: Pre-release versions (e.g., `1.2.0-beta`) are older than release versions (e.g., `1.2.0`)

Solution: Use release versions in `minLauncherInitialAppVersion`

Frequent Reinstalls Required

Problem: Users must reinstall too often

Solution:

- Use more conservative version requirements
- Allow time between launcher-requiring updates
- Consider if the launcher update is truly necessary

Branch Releases Don't Enforce

Symptom: Branch-based releases ignore `minLauncherInitialAppVersion`

Explanation: This is intentional to support rapid development and testing

Solution: Use version-based releases for launcher enforcement

Resources

- **Official Website**: https://www.jdeploy.com/
- **Download**: https://github.com/shannah/jdeploy-desktop-gui/releases/latest
- **Documentation**: https://www.jdeploy.com/docs/manual/
- **GitHub Repository**: https://github.com/shannah/jdeploy
- **Support**: GitHub Issues and Discussions
- **RFC Document**: Launcher Version Enforcement RFC