InstallAnywhere 2024 R2 Release Notes

Originally released July 2024; updated to include R2 (January 2025)

Introduction

InstallAnywhere is the leading multiplatform installation and container development solution for application producers who need to deliver a professional and consistent installation experience for physical, virtual, and cloud environments as well as container deployments.

Changes in InstallAnywhere 2024 R2

InstallAnywhere 2024 R2 includes the following new feature:

- Introduced Support for ARM64

Introduced Support for ARM64

InstallAnywhere 2024 R2 introduces support for ARM64 (AARCH64) Linux distribution, which enables you to build projects on both Windows and Linux operating systems running the ARM64 (AARCH64) architecture.

Note ▪ *This change was tracked in IA-21776.*

New Features in InstallAnywhere 2024 R1

InstallAnywhere 2024 R1 includes the following new features:

- Ability to Configure Custom Signing Solution for Digital Signing
- Introduced New Component for Custom Code Panels
- Introduced Exit Code for Invalid Java Virtual Machine

Ability to Configure Custom Signing Solution for Digital Signing

InstallAnywhere 2024 R1 introduces new signing type settings that enable you to select and configure a custom signing solution to digitally sign your Windows-based installers (installer .exe file, installer launcher, and uninstaller launcher) during build time. If the user prefers to use the custom signing solution instead of the InstallAnywhere standard solution, these settings are essential.

Selecting the **Custom** signing type enables additional fields where a custom signing tool path and arguments can be configured. On the **Project > Platforms > Windows** area, under the **Digital Signing** setting, you can use following new settings to configure the custom signing solution:

- **Signing Method**—This setting enables you to specify the method to digitally sign your Windows-based installers during build time. The available options for this setting are:

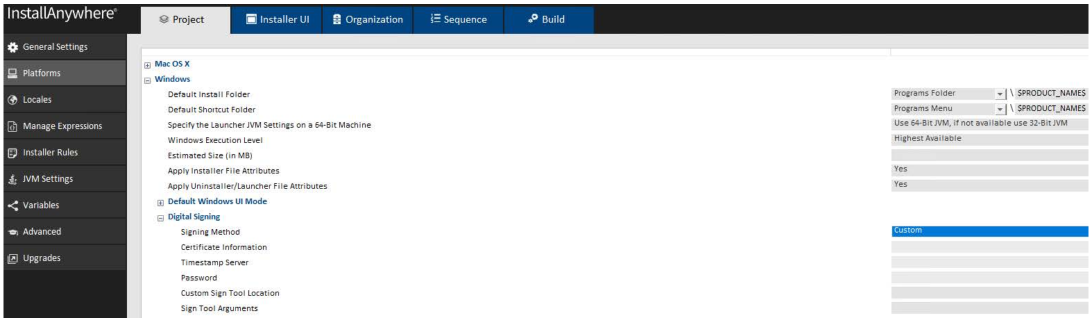
- **Standard**—Select this option to utilize the default InstallAnywhere sign tool to digitally sign your Windows-based installers during build time.
- **Custom**—Select this option to use an external sign tool to digitally sign your Windows-based installers during build time. Selecting this option enables the **Custom Sign Tool Location** and **Sign Tool Arguments** settings.

By default, the **Signing Method** setting is set to the **Standard** option.

- **Custom Sign Tool Location**—Specify the location of the external sign tool that you want to use to digitally sign your Windows-based installers during build time. To specify the sign tool's location, click the ellipsis button (...).
- **Sign Tool Arguments**—Specify the valid argument for an external sign tool’s configuration. For instance, the following argument can be used for an external sign tool’s configuration:

```
sign /a /fd SHA256 /t http://timestamp.digicert.com
```

The following displays the enabled **Custom Sign Tool Location** and **Sign Tool Arguments** settings when the **Signing Method** setting is set to the **Custom** option:



⊞ Mac OS X	
⊟ Windows	
Default Install Folder	Programs Folder \ $PRODUCT_NAME$
Default Shortcut Folder	Programs Menu \ $PRODUCT_NAME$
Specify the Launcher JVM Settings on a 64-Bit Machine	Use 64-Bit JVM, if not available use 32-Bit JVM
Windows Execution Level	Highest Available
Estimated Size (in MB)	
Apply Installer File Attributes	Yes
Apply Uninstaller/Launcher File Attributes	Yes
⊞ Default Windows UI Mode	
⊟ Digital Signing	
Signing Method	Custom
Certificate Information	
Timestamp Server	
Password	
Custom Sign Tool Location	
Sign Tool Arguments	

Figure 1: Enabled Settings for Custom Signing

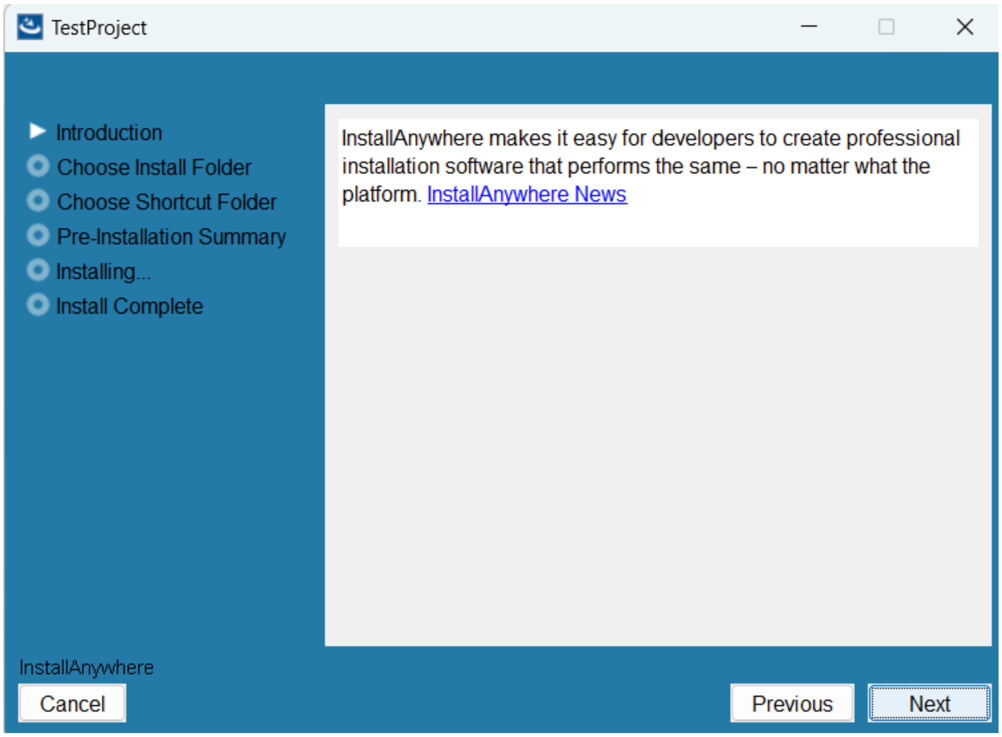
Note *▪ This change was tracked in IA-21714.*

Introduced New Component for Custom Code Panels

Previously, InstallAnywhere Custom Code panels were unable to support multiple-line contents and URLs.

InstallAnywhere 2024 R1 introduces a new component named, `IATextPane`, which enables the custom code panels to support multiple-line contents, URLs, and DPI scaling. This component is available in the `com.zerog.ia.api.pub.controls` package and can be used in Custom Code panels. The Javadocs for this component are available in the `\javadoc\com\zerog\ia\api\pub\controls` directory.

The following displays a Custom Code panel using the `IATextPane` component during installation:



InstallAnywhere makes it easy for developers to create professional installation software that performs the same – no matter what the platform. InstallAnywhere News

Figure 2: Custom Code Panel using `IATextPane`

Note ▪ *This change was tracked in IA-21448.*

Introduced Exit Code for Invalid Java Virtual Machine

Previously, when an InstallAnywhere installer failed to detect a valid Java Virtual Machine (JVM) during installation, the following error message appeared without an exit code:

```
Could not find a valid Java Virtual machine to load. You may need to reinstall a supported java
virtual machine.
```

In InstallAnywhere 2024 R1, if an InstallAnywhere installer fails to detect a valid Java Virtual Machine (JVM) during installation, an exit code `106` will be returned and will be displayed on the console along with the error message as:

```
Could not find a valid Java Virtual machine to load. You may need to reinstall a supported java
virtual machine. exit code : 106
```

Note ▪ *This change was tracked in IA-21480.*

Enhancements

InstallAnywhere 2024 includes the following enhancements:

- Enhancements in InstallAnywhere 2024 R2
- Enhancements in InstallAnywhere 2024 R1

Enhancements in InstallAnywhere 2024 R2

This section lists enhancements that were included in InstallAnywhere 2024 R2:

- Custom Code API Methods to Retrieve Font and Foreground Color Used in Standard Panel
- Ability to Restrict Non-Platform-Specific Service Support Files Build into Installers

Custom Code API Methods to Retrieve Font and Foreground Color Used in Standard Panel

In previous releases, users were unable to retrieve the font and foreground color used in the standard InstallAnywhere panel from within a custom code panel.

InstallAnywhere 2024 R2 enables you to retrieve the font and foreground color used in the standard InstallAnywhere panel from within the custom code panel by introducing the following custom code API methods:

- **public Font getInstallerFont()**—Use this API method to retrieve the font used in the standard InstallAnywhere panel from within the custom code panel.
- **public Color getInstallerFontColor()**—Use this API method to retrieve the foreground color used in the standard InstallAnywhere panel from within the custom code panel.

You can use these methods on Java Swing components, which is used in the custom code panel, through the GUIAccess service as follows:

```
GUIAccess gui = (GUIAccess) proxy.getService(GUIAccess.class);
<swing component>.setFont(gui.getInstallerFont());
<swing component>.setForeground(gui.getInstallerFontColor());
```

Note ▪ *This change was tracked in IA-19831.*

Ability to Restrict Non-Platform-Specific Service Support Files Build into Installers

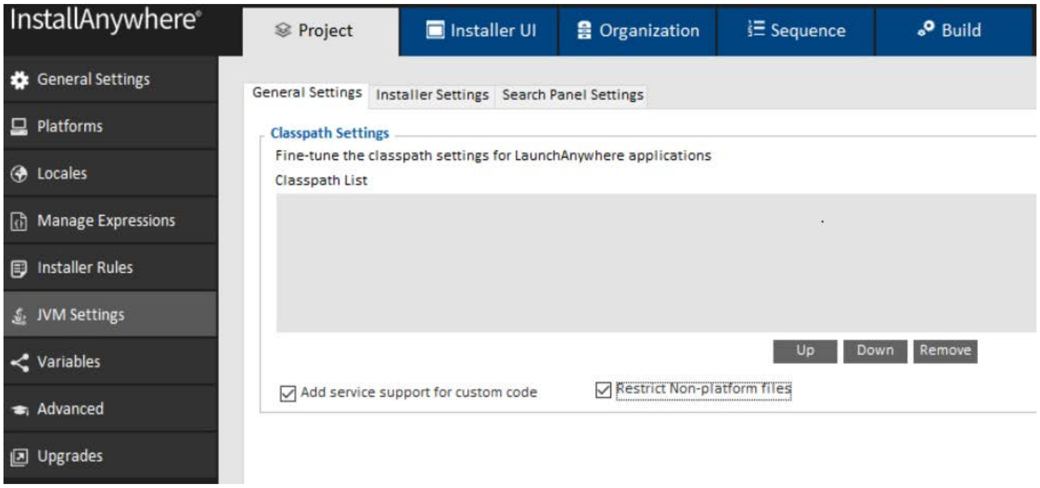
Previously, when attempting to include support for the custom code services layer, both platform-specific and non-platform-specific service support files were built into each target installer and uninstaller archive.

In InstallAnywhere 2024 R2, a new check box, **Restrict Non-platform files**, has been introduced on the **General Settings** tab in the **JVM Settings** view of the **Project** page. This check box allows you to build only the platform-specific service support files into the installer and uninstaller archives, and restrict the non-platform-specific service support files, when including support for the custom code services layer. You can use this check box to control the build of service support files in the installer and uninstaller archives as follows:

- Selecting this check box allows only the platform-specific service support files from the `<IA_HOME>/resource/services/ppk` directory to build into the installer and uninstaller archives for each target platform.
- Clearing this check box allows both platform-specific and non-platform-specific service support files from the `<IA_HOME>/resource/services/ppk` directory to build into the installer and uninstaller archives.

  By default, this check box is cleared.

The following shows the selected **Restrict Non-platform files** check box on the **General Settings** tab in the **JVM Settings** view:



Classpath Settings

Fine-tune the classpath settings for LaunchAnywhere applications

Classpath List

☑ Add service support for custom code

☑ Restrict Non-platform files

Figure 3: Selected Restrict Non-platform files Check Box

***Note** ▪ This change was tracked in IA-21488.*

Enhancements in InstallAnywhere 2024 R1

This section lists enhancements that were included in InstallAnywhere 2024 R1:

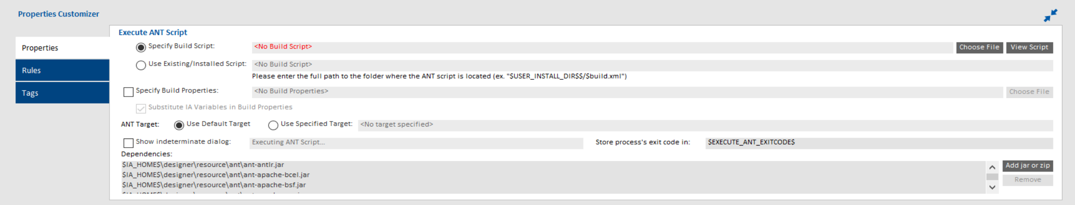
- New Execute Ant Script Action’s Option to Store Exit Code
- Included Password Variable Value in installvariables.properties
- Appropriate Error Message for Invalid Java Virtual Machine
- New Command-Line Argument to Set the Temp Directory
- Enhanced the Installer Launch in Console Mode

New Execute Ant Script Action’s Option to Store Exit Code

In previous releases, an InstallAnywhere installer was unable to store the returned value or exit code from an Ant script during installation.

InstallAnywhere 2024 R1 introduces a new option named **Store process's exit code in** on the **Properties** tab in the **Properties Customizer** area of the **Execute ANT Script: <No Build Script>** action (a **General** action) for any selected view on the **Sequence** page. This option enables you to specify a variable name which is used to store the value or exit code returned from an Ant script during installation.

You can specify any customized variable name. By default, the variable name is set to **$EXECUTE_ANT_EXITCODE$**. The following shows the **Store process's exit code in** option with the default variable name:



● Specify Build Script: <No Build Script>

○ Use Existing/Installed Script: <No Build Script>

Please enter the full path to the folder where the ANT script is located (ex. "$USER_INSTALL_DIR$$/$build.xml")

□ Specify Build Properties: <No Build Properties>

☑ Substitute IA Variables in Build Properties

ANT Target: ● Use Default Target ○ Use Specified Target: <No target specified>

□ Show indeterminate dialog: Executing ANT Script...

Store process's exit code in: $EXECUTE_ANT_EXITCODE$

Dependencies:

IA_HOME\designer\resource\ant\ant-antlr.jar
IA_HOME\designer\resource\ant\ant-apache-bcel.jar
IA_HOME\designer\resource\ant\ant-apache-bsf.jar

Figure 4: Store process's exit code in Option

Note ▪ *By default, an exit code or value is not returned by the Ant script. To return a value or exit code, the variable name specified in this option, must be included in the ANT script by either using the Exec tag with the `resultproperty` attribute or by using the Property tag with the name attribute.*

*For instance, if the $EXECUTE_ANT_EXITCODE$ variable name is specified in the **Store process's exit code in** option, the ANT script includes the $EXECUTE_ANT_EXITCODE$ variable name either by using `failonerror="true"resultproperty="EXECUTE_ANT_EXITCODE"` in the Exec tag or by using <property `name="EXECUTE_ANT_EXITCODE" value="1" />`.*

Note ▪ *This change was tracked in IA-4927.*

Included Password Variable Value in installvariables.properties

In previous releases, when the password variable—usually configured for encrypted value using the **Configure Variables** area in the **Variables** view on the **Project** page—was specified through the **Console: Get Password: Enter Password** or **Panel: Get Password: Enter Password** action in the selected view on the **Sequence** page, the `installvariables.properties` file included only the password variable, not its encrypted value.

In InstallAnywhere 2024 R1, when the password variable is specified through the **Console: Get Password: Enter Password** or **Panel: Get Password: Enter Password** action in the selected view on the **Sequence** page, the `installvariables.properties` file includes both the password variable and its encrypted value, which is accessible during uninstallation.

***Note* ▪ This change was tracked in IA-20045.**

Appropriate Error Message for Invalid Java Virtual Machine

Previously, the following error appeared on the installer when there was a failure to locate a valid Java virtual machine (JVM) during installation or uninstallation:

```
No Java virtual machine could be found from your PATH environment variable. You must install a
VM prior to running this program.
```

In InstallAnywhere 2024 R1, when the installer fails to locate a valid Java virtual machine (JVM) during installation or uninstallation then an appropriate error message appears:

```
No Valid Java virtual machine could be found from your PATH environment variable. You must
install a valid VM prior to running this program. Required Java VM : 11.0+. appears.
```

Note ▪ *11.0+ is a dynamic value, which changes with the value set for **Valid VM list** setting in the **Installer Settings** tab on the **JVM Settings** view of the **project** page.*

Note ▪ *This change was tracked in IA-21553.*

New Command-Line Argument to Set the Temp Directory

InstallAnywhere 2024 R1 introduced a new command-line argument named, `-tempdir`, that enables you to set the temp directory path where the InstallAnywhere launchers are required to be extracted. To set the temp directory path, you can enter the following at the command line:

```
<installer_name/uninstaller_name> -tempdir "path"
```

You can set the required temp directory path by using the `-tempdir` argument with the installer or uninstaller name.

Note ▪ *A valid directory path must be specified at the command line. if the specified directory path is not accessible or is not a valid path, then the default temp directory path will be utilized by the InstallAnywhere installer/uninstaller.*

For more information, see Using Command-Line Arguments with Installers and Uninstallers in the InstallAnywhere Help Book.

Note ▪ *This change was tracked in IA-20497.*

Enhanced the Installer Launch in Console Mode

Previously, an InstallAnywhere installer exited from the installation without an appropriate error message when it failed to find Java on the machine during a launch in the console mode.

In InstallAnywhere 2024 R1, if an InstallAnywhere installer launches in the console mode and Java is not detected on the machine then an appropriate error message is displayed on the console and remains on the console until the Enter key is used.

***Note** ▪ This enhancement is applicable for both the Windows and macOS platforms.*

***Note** ▪ This Change was tracked in IA-21665.*

Important Information

Note the following important information regarding InstallAnywhere 2024:

- Evaluating InstallAnywhere
- Obtaining the Installation and License Files for InstallAnywhere

Evaluating InstallAnywhere

Note that if you have not purchased a license for InstallAnywhere, you can install it and use it for a limited number of days without setting up the licensing. When you are using InstallAnywhere in this scenario, it operates in evaluation mode. The licensing wizard that InstallAnywhere displays whenever you launch InstallAnywhere in evaluation mode shows you how many days are left in the evaluation period. If you do not set up the licensing within the evaluation period, InstallAnywhere stops working when the evaluation period ends. You can set up licensing at any time before or after the evaluation period ends.

When you use InstallAnywhere in evaluation mode, there is a time limit for running installers that it creates. If you build an installer in the evaluation version of InstallAnywhere, your installer will stop working after 3 days.

Obtaining the Installation and License Files for InstallAnywhere

You can obtain the installation and license files for InstallAnywhere through the Revenera Product and License Center. For instructions, see the Download and licensing instructions for InstallAnywhere. If you purchased concurrent licenses of InstallAnywhere, the license server software is also available for download from that same site.

Resolved Issues

This section lists the customer issues that were resolved in the following version of InstallAnywhere 2024:

- InstallAnywhere 2024 R2
- InstallAnywhere 2024 R1

InstallAnywhere 2024 R2

The following issues have been fixed in InstallAnywhere 2024 R2:

Issue Number	Issue Summary
IA-21777	A failure was observed while attempting to build an InstallAnywhere project using win64 or redhat64 as the build configuration name. This issue has been fixed.
IA-21754	Using multiple Read/Modify XML File actions to replace several tag values led to the addition of multiple blank spaces and modifications to the indentations in the XML file. This issue has been fixed.
IA-21904	An installer launched in console mode with the Japanese locale failed to display Japanese strings; instead it displayed English strings. This issue has been fixed.
IA-20332	Configuring the **Control Properties** for a Choice Group component in the **Get User Input - Advanced Panel** action with a custom font led to the font being displayed with improper scale on a custom panel during build. This issue has been fixed.
IA-21779	A string search failure was observed in an InstallAnywhere project; a `java.lang.NullPointerException` message was displayed on the debug console. This issue has been fixed.
IA-21149	InstallAnywhere executed a malicious executable file renamed as `remove.exe`—located in the `temp` directory—with administrator privileges on the Windows platform. This execution occurred with the same elevated privileges as the `Notepad.exe` file execution by an elevated user. This issue has been fixed.
IA-21736	An installer step label failed to display the custom value configured in a `custom_<locale>` file of an InstallAnywhere installer. Instead, the custom value reverted the default value in the `custom_<locale>` file during build. This issue has been fixed.

Issue Number	Issue Summary
IA-21470	Attempting to build the helper tool using the macOS SDK 14 led to a build failure with the following error message: `/Applications/InstallAnywhere 2023/resource/nativetools/macosx/ht-signer/authentication/Authenticator.xcodeproj: warning: The macOS deployment target 'MACOSX_DEPLOYMENT_TARGET' is set to 10.8, but the range of supported deployment target versions is 10.13 to 14.0.99. (in target 'com.flexera.ia.helper' from project 'Authenticator')` This issue has been fixed.
IA-21418	When an installer was launched in maintenance mode with the **Add Features** option, all features appeared as selected when navigating back to the **Choose Install Set** panel, even though not all features were originally selected. This issue has been fixed.
IA-21902	The `ObjectID` of the `ManageExpressions` element in an InstallAnywhere project was updated when the InstallAnywhere project was loaded and saved, even though no modifications were made to that project. This issue has been fixed.
IA-20929 **IA-21771**	Notarization of installer failed when selecting the **Add service support for custom code** check box in the **General Settings** tab on the **JVM Settings** view of the **Project** page or including the files located in the `IA_HOME/resource/services/ppk` directory as payload. This issue has been fixed.
IA-21840	A failure was observed when attempting to sign and notarize an installer via the `SignInstallerXcode13Above.sh` script. This issue has been fixed.
IA-21953	Launching an InstallAnywhere installer with the Adopt JDK 21 VM pack led the Get Password console panel to display a prompt message twice, as follows: `Please Enter the Password: Please Enter the Password:*` This issue has been fixed.
IA-21770	Using the **Modify Text File - Multiple Files** action to search and replace strings in an XML file that included an empty last line resulted in the removal of the same line during installation. This issue has been fixed.
IA-21973	The addition of the `lax.nl.java.option.additional` LAX property with a value in the LaunchAnywhere properties caused a blank entry to appear in the `info.plist` file, resulting in the failure of the application launcher on macOS. This issue has been fixed.
IA-21971	For a migrated customized project, clicking the **Custom UI Designer** button in the **UI Panel Settings** area failed to open the themes dialog box and displayed an exception. This issue has been fixed.

Issue Number	Issue Summary
IA-20584	A failure was observed on the macOS platform when attempting to install the LaunchAnywhere launcher with the `lax.nl.java.option.additional` property set to **`--illegal-access=permit`**. This issue has been fixed.
IA-21876	The LAX executable file failed to redirect the stdout output to a file. This issue has been fixed.
IA-22012	Using the Choose Java VM panel in the Japanese locale displayed the same Java path multiple times when navigating using the **Previous** and **Next** buttons in the installation workflow. This issue has been fixed.
IA-21988	Multiple vulnerabilities were detected for the Amazon Corretto Java versions prior to 11.0.24.8.1, which were installed with InstallAnywhere. This issue has been fixed.
IA-21828	The Display Message panel failed to display the value of *`JDK_HOME`* variable during installation on machines running Java versions higher than Java 8. This issue has been fixed.
IA-21104	Adding a new build configuration to a project that supported only German locale caused that build configuration to support only English and also exceptions were generated while running the built installer. This issue has been fixed.
IA-22013	Using Java 21 or later led the **Cancel** button in the Install Progress panel to appear grayed out and become unresponsive when clicked. This issue has been fixed.
IA-22035	The maximum heap size defined by the `lax.nl.java.option.java.heap.size.max` property in a Windows installer was limited to 2147483647. This Issue has been fixed.

InstallAnywhere 2024 R1

The following issues have been fixed in InstallAnywhere 2024 R1:

Issue Number	Issue Summary
IA-15956	On the Linux platform, setting the *`IATEMPDIR`* variable failed to disable an InstallAnywhere installer from using the system temporary directory for some activities. This issue has been fixed.
IA-21340	Navigating through the Pre-Install sequence led the installer to hang and display the `StackOverflowError` on the console. This issue has been fixed.

Issue Number	Issue Summary
IA-20888	The **Create JRE VM Pack Wizard** failed to create a VM pack from an extracted macOS JRE since Java 9 and displayed the `Please choose a valid path for jre` error message. This issue has been fixed.
IA-20644	An installer corrupted the *`$USER_INSTALL_DIR$`* variable by deleting the files and failed to extract additional files via the **Expand Archive (7-zip)** action. This issue has been fixed.
IA-21575	If a user name ended with an exclamation mark, the installer would fail to launch with the `Couldn't find the InstallScript to run the installer from! Unable to load the script.` error message. This issue has been fixed.
IA-21059	An installer with the **Extract Archive** action was failed to retrieve source files' timestamp from a .zip archive on a Linux machine, which resulted in the display of **File OverWrite** prompts for each file extracted from the archive. This issue has been fixed.
IA-21647	InstallAnywhere failed to pick up Java files from the **Java Home** folder and instead picked up Java files from the **Internet Plug In** folder, which led to an unsuccessful application installation. This issue has been fixed.
IA-21646	When the custom icon of an uninstaller was changed, the uninstaller application launch failure was observed on macOS Ventura/Sonoma with the following message: `[Name of the app] is damaged and can't be opened. You should move it to the Trash.` This issue has been fixed.
IA-21177	The **Expand Archive (7-zip)** action failed to extract a file to the directory specified by the *`IATEMPDIR`* environment variable. This issue has been fixed.
IA-21690	A crash was observed with `1067` error, when using Java 21 and later versions and launching a console launcher registered as Windows Services. This issue has been fixed.
IA-21413	InstallAnywhere installer failed to launch in silent mode on a target machine that was running in the Hebrew locale. This issue has been fixed.
IA-21607	When two duplicate files were included in a project, the installer installed the one file at the correct size and the other at 0 bytes. This issue has been fixed.
IA-21586	Japanese characters that were copied to the InstallAnywhere Graphical User Interface (GUI) led to the display of unknown characters. This issue has been fixed.
IA-21351	Files in the SpeedFolder with a defined filter failed to install in the Add feature type of maintenance mode. This issue has been fixed.

Issue Number	Issue Summary
IA-16850	Inadequate permission for the /tmp directory usage and the InstallAnywhere installer's failure to effectively utilize the directory specified by the *IATEMPDIR* environment variable, led to an installation failure with "unlicensed version" error message. This issue has been fixed.
IA-21375	The gradle process usage in the jar files signing led the installer to be installed with errors. This issue has been fixed.
IA-21460	A failure was observed when attempting to accept the Enter key as the default option during console mode uninstallation with the maintenance mode enabled. This issue has been fixed.
IA-21649	When a machine was connected to a LAN, InstallAnywhere generated the license file based on the LAN adapter MacID instead of the machine MacID. This issue has been fixed.
IA-21212	A build appliance failed and displayed the `Message: Credential Information is specified, but it is incomplete.` error message. This issue has been fixed.
IA-15766	A signed custom code action failed during runtime due to the addition of more than 20 dependencies. This issue has been fixed.
IA-21622	An installer failed to install the files and folders in the expected location via the **Install from Manifest** action. This issue has been fixed.
IA-21642	A failure was observed on the links usage that are included in the product documentation to access the javadocs. This issue has been fixed.
IA-21573	The **Other Java-Enabled platforms** build target customizer based installer failed to launch with Java 21 and led to display the following error message: `Exception in thread "main" java.lang.NoClassDefFoundError: java/lang/Compiler at install.` This issue has been fixed.
IA-21473	A failure was observed while testing the Oracle database connection on the Oracle 19C. This issue has been fixed.
IA-21650	The files were failed to appear in the JavaDoc folder. This issue has been fixed.
IA-21620	The Java vendor name watermark was appeared during console mode Installation. This has been removed.
IA-12257	Using the **Set Windows Registry - Multiple Entries** and **Set Windows Registry - Single Entry** actions failed to revert the DWORD values. This issue has been fixed.

Issue Number	Issue Summary
IA-21299	The Password field in the response file was recorded empty after silent installation. This issue has been fixed.
IA-21621	The `lax.nl.java.compiler` LAX property was listed in the uninstaller lax file. This has been removed
IA-21014	When a customized installer title image icon included source path variables, a Merge module project failed to import to the parent project with the `IAScript.loadScript()` error. This issue has been fixed.

System Requirements

The following are the system requirements for InstallAnywhere 2024:

- Requirements for Systems that are Running InstallAnywhere (Authoring Environment)
- Requirements for Target Systems (Installer Run-Time Environment)
- Supported Java Virtual Machines

Requirements for Systems that are Running InstallAnywhere (Authoring Environment)

RAM

256 MB; 512 MB preferred

Hard Disk Free Space

500 MB

Color

High color (16-bit color depth)

Resolution

Minimum 1200 x 800

Operating System

InstallAnywhere runs on the latest versions of these operating systems, fully updated with the most recent patches and service packs.

Operating System	Supported Versions
Windows	● Windows 11 ● Windows 10 ● Windows Server 2012 R2, 2016, 2019, 2022, and 2025 ● Windows 7 and 8.1 (x86 and x64)
Apple	● macOS Sequoia 15.1.1 with Oracle and Amazon Corretto Java 11.0.15+ ● macOS Sonoma (14) with Oracle and Amazon Corretto Java 11.0.15+ ● macOS Ventura (13) with Oracle and Amazon Corretto Java 11.0.15+ ● macOS Ventura (13) with Oracle and Amazon Corretto Java 11.0.15+ on Apple M1 ● macOS Monterey (12) with Oracle and Amazon Corretto Java 11.0.7+ on Apple M1 ● macOS Big Sur (11) with Oracle and Amazon Corretto Java 11.0.7+ ● macOS Big Sur (11) with Oracle and Amazon Corretto Java 11.0.7+ on Apple M1
Linux	● Red Hat Enterprise Linux 7.9, 8 (x64), 8.3, 8.4, 8.5, 8.6, 8.7, 9.0, 9.1, 9.3, and 9.4 ● OpenSUSE Leap 42.3 (x64) and 15.6 ● OpenSUSE Linux 15.3, 15.4, and 15.6 ● SUSE Linux Enterprise 15 ● Linux PPC 64-bit (build time only) only with Java 11 ● Ubuntu 21.4, 22.04, 22.10, 23.10, and 24.04 ● Fedora 34, 36, 37, 39, and 40 ● CentOS Stream 9.0 **Note ▪** *When you install InstallAnywhere on a Linux system, the installation attempts to create a symbolic link to the default Linux loader (/lib/ld-linux.so.2) if a link with the same name is not already present. The symbolic link is necessary for the host ID to be displayed on the Host ID dialog, and it is also necessary for successful node-locked licensing. The link is present on systems that are Linux Standard Base (LSB) 3 compliant, but it may not be present on systems that are not LSB compliant. For more information, see Knowledge Base article Q209204.*

Installers can be built from any supported authoring platform for any other supported target platform or language. Localizations for 32 languages are included.

***Note* ▪ *InstallAnywhere installers no longer distrubute Java 8. InstallAnywhere 2024 supports only Java 11 and later versions. The JAR files in Java 8, usually present in the* `jre/lib/` *and* `jre/lib/ext/` *directories, are also not distributed with Java 11 and later versions, therefore these files will be unavailable*.

Requirements for Target Systems (Installer Run-Time Environment)

RAM

64 MB

Color

High color (16-bit color depth)

Resolution

Minimum 640 x 480

Operating System

Installers run on any version of these operating systems, as long as the operating system supports Java 11 or 17. InstallAnywhere-generated installers are not supported on beta versions or on early-access releases unless they are explicitly mentioned.

Operating System	Supported Versions
Windows	● Windows 11 ● Windows 11 ARM64 (AARCH64) ● Windows 10 ● Windows Server 2012 R2, 2016, 2019, 2022, and 2025 ● Windows 7 and 8.1 (x86 and x64) ***Note* ▪ *Windows-based target systems must also support the SSE2 instruction set.***

Operating System	Supported Versions
Apple	● macOS Sequoia 15.1.1 with Oracle and Amazon Corretto Java 11.0.15+ ● macOS Sonoma (14) with Oracle and Amazon Corretto Java 11.0.15+ ● macOS Ventura (13) with Oracle and Amazon Corretto Java 11.0.15+ ● macOS Ventura (13) with Oracle and Amazon Corretto Java 11.0.15+ on Apple M1 ● macOS Monterey (12) with Oracle and Amazon Corretto Java 11.0.7+, Java 17 on Apple M1 ● macOS Big Sur (11) with Oracle and Amazon Corretto Java 11.0.7+, Java 17 ● macOS Big Sur (11) with Oracle and Amazon Corretto Java 11.0.7+, Java 17 on Apple M1 ● macOS Catalina (10.15) with Oracle Java 11.0.7+ ● macOS Mojave (10.14) with Oracle Java 11.0.7+
Linux	● CentOS Stream 9.0 and CentOS Stream 9 ARM64 (AARCH64) ● Red Hat Enterprise Linux 7.9, 8 (x64), 8.3, 8.4, 8.5, 8.6, 8.7, 9.0, 9.1, 9.3, and 9.4 ● Red Hat Enterprise Linux 7.2 and 8 for PowerPC (little endian) ● Red Hat Linux 7.1 and 8 for PowerPC (little endian - silent and console mode only) ● OpenSUSE Linux 15.3, 15.4, and 15.6 ● Oracle Linux 8.5, 9, 9.3, and 9.4 ● SUSE Linux Enterprise 15 ● Ubuntu 21.4, 22.04, 22.10, 23.10, and 24.04 ● Fedora 34, 36, 37, 39, and 40
Solaris	● Solaris 11 (x86 and SPARC) ● Solaris 9, 10 (x86, SPARC, and AMD-64) ● HP-UX
HP-UX	● HP-UX 11i (Itanium 2 and PA-RISC)
AIX	● AIX 7.1, 7.2, and 7.3 (Power/PowerPC)
IBM	● i5/OS (OS/400) on System i - V5R3 and V5R4 (Enterprise Edition only), IBM i 6.1, and IBM i 7.1 ● z/OS

Operating System	Supported Versions
Other	• FreeBSD • Other Linux and UNIX operating systems (POSIX-compliant shell required)

Note *▪ InstallAnywhere installers no longer distrubute Java 8. InstallAnywhere 2024 supports only Java 11 and later versions. The JAR files in Java 8, usually present in the `jre/lib/` and `jre/lib/ext/` directories, are also not distributed with Java 11 and later versions, therefore these files will be unavailable*.

Supported Java Virtual Machines

InstallAnywhere supports the following Java virtual machines:

Manufacturer	Supported Versions
IBM	11, 17, and 21
HP	11.0.2
Sun/Oracle	11, 11.0.7, 12, 14.0.1, 15, 17, 18, 19, 20, 21, 22, and 23
OpenJDK	11, 11.0.7, 12, 14.0.1, 15, 17, 18, 19, 20, 21, 22, and 23
Amazon Corretto	11, 11.0.7, 15, 17, 18, 19, 20, 21, 22, and 23
Adoptium/AdoptOpenJDK	11.0.7, 15, 16, 17, 21, 22, and 23
Azul Platform Core JRE	11, 17, 18, 19, 20, 21, 22, and 23

The InstallAnywhere installer installs OpenJDK11.0.2. Any Java virtual machine can be bundled with an installer ensuring that the target system meets the minimum requirements for both the installers and your applications. To download additional JRE VM packs, visit https://www.revenera.com/install/products/installanywhere/installanywhere-files-utilities.html and click the VM Packs option.

InstallAnywhere-generated installers are not supported on beta versions or on early-access releases of Java.

InstallAnywhere installers no longer distribute Java 8. InstallAnywhere 2024 supports only Java 11 and later versions. The JAR files in Java 8, usually present in the `jre/lib/` and `jre/lib/ext/` directories, are also not distributed with Java 11 and later versions, therefore these files will be unavailable.

Known Issues

There is no known issue in InstallAnywhere 2024.

Legal Information

Copyright Notice

Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see https://www.revenera.com/legal/intellectual-property.html. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.