.NET 6.0

Release Notes for .NET 6.0 RPM packages

Last Updated: 2023-08-24

Legal Notice

Abstract

The Release Notes for .NET 6.0 RPM packages provide high-level coverage of the features and functionality that comprise the .NET 6.0 platform and document known problems in this release.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 4

CHAPTER 1. AVAILABILITY 5

CHAPTER 2. OVERVIEW 6

CHAPTER 3. FEATURES AND BENEFITS 7

3.1. CURRENT FEATURES AND BENEFITS 7

3.2. NEW FEATURES AND BENEFITS 7

CHAPTER 4. SUPPORTED OPERATING SYSTEMS AND ARCHITECTURES 9

CHAPTER 5. CUSTOMER PRIVACY 10

CHAPTER 6. SUPPORT 11

6.1. CONTACT OPTIONS 11

6.2. FREQUENTLY ASKED QUESTIONS 11

6.3. ADDITIONAL SUPPORT RESOURCES 12

CHAPTER 7. KNOWN ISSUES 13

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the Bugzilla website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

CHAPTER 1. AVAILABILITY

Red Hat provides a distribution of .NET that enables developers to create applications using the C#, Visual Basic, and F# languages and then deploy them on Red Hat Enterprise Linux (RHEL), Red Hat OpenShift Container Platform, or other platforms. A no-cost Red Hat Enterprise Linux Developer Subscription is available, including a full suite of tools for container development.

- For Red Hat Enterprise Linux 7, .NET 6.0 is available as the RPM **rh-dotnet60** in the following repositories:

    - RHEL 7 Server: **rhel-7-server-dotnet-rpms**
    - RHEL 7 Workstation: **rhel-7-workstation-dotnet-rpms**
    - RHEL 7 HPC: **rhel-7-hpc-node-dotnet-rpms**

    Full instructions for installing .NET 6.0 on Red Hat Enterprise Linux 7 are available in the Getting started with .NET on RHEL 7 guide.

IMPORTANT

.NET on Red Hat Enterprise Linux 7 will no longer receive support or updates after **30 June 2024**. As such, Red Hat recommends either using RHEL 8-based .NET 6.0 containers on RHEL 7 after **30 June 2024** or upgrading to a more recent version of RHEL.

- For Red Hat Enterprise Linux 8, .NET 6.0 is available as the RPM **dotnet-sdk-6.0** in the AppStream repositories. The AppStream repositories are enabled by default in RHEL 8.

- .NET 6.0 is available for **aarch64**, **s390x**, and **x86_64** architectures on Red Hat Enterprise Linux 8 and OpenShift Container Platform.
  Full instructions for installing .NET 6.0 on RHEL 8 are available in the Getting started with .NET on RHEL 8 guide.

CHAPTER 2. OVERVIEW

.NET is a general purpose, modular, cross-platform, and open source implementation of .NET that features automatic memory management and modern programming languages. It allows users to build high-quality applications efficiently. .NET is available on RHEL 7, RHEL 8, RHEL 9.

.NET 6.0 is a Long Term Support (LTS) release. LTS releases are generally supported for around 3 years. For more information, see the Life Cycle and Support Policies for the .NET Program .

.NET offers:

- The ability to follow a microservices-based approach, where some components are built with .NET and others with Java, but all can run on a common, supported platform in RHEL.
- The capacity to more easily develop new .NET workloads on Microsoft Windows. You can deploy and run on either RHEL or Windows Server.
- A heterogeneous data center, where the underlying infrastructure is capable of running .NET applications without having to rely solely on Windows Server.

CHAPTER 3. FEATURES AND BENEFITS

3.1. CURRENT FEATURES AND BENEFITS

.NET 6.0 offers the following features and benefits.

- Runtime and framework libraries
  .NET consists of the runtime and the framework libraries as well as compilers, build tools, tools to fetch NuGet packages, and a command-line interface to tie everything together. Benefits include:
  - Automatic memory management
  - Type safety
  - Delegates and lambdas
  - Generic types
  - Language Integrated Query (LINQ)
  - Async programming
  - Native interoperability
  - Source generators
- .NET 6.0 supports developing applications using ASP.NET Core 6.0 and EF Core 6.0, which bring benefits such as:
  - Lightweight and modular HTTP request pipeline
  - Ability to host on a web server or self-host in your own process
  - Built on .NET, which supports true side-by-side app versioning
  - Integrated support for creating and using NuGet packages
  - Single aligned web stack for web UI and web APIs
  - Cloud-ready environment-based configuration
  - Built-in support for dependency injection
  - Tools that simplify modern web development

3.2. NEW FEATURES AND BENEFITS

Support for 64-bit Arm (aarch 64), IBM Z, and LinuxONE (s390x)

- .NET 6.0 introduces support for 64-bit Arm running on Red Hat Enterprise Linux 8.
- .NET 6.0 introduces support for IBM Z and LinuxONE running on Red Hat Enterprise Linux 8 and OpenShift Container Platform 4.2 or later.

.NET 6.0 continues to broaden its support and tools for application development in an open source environment. The latest version of .NET includes the following improvements:

- Support for C# 10
- Support for F# 6
- Single-file source programs
- Performance improvements in base libraries, GC and JIT
- Source-generators for logging and JSON
- Better diagnostics with dotnet-monitor

CHAPTER 4. SUPPORTED OPERATING SYSTEMS AND ARCHITECTURES

.NET 6.0 is available on **x86_64**, **aarch64**, and **s390x** on Red Hat Enterprise Linux 8 and later.

Table 4.1. Supported deployment environments for .NET 6.0

Platform	Architecture	RPM	Repository
Red Hat Enterprise Linux 7 (Until **30 June 2024**)	AMD64 and Intel 64 (**x86_64**)	rh-dotnet60	Red Hat Enterprise Linux 7 Server: rhel-7-server-dotnet-rpms Red Hat Enterprise Linux 7 Workstation: rhel-7-workstation-dotnet-rpm Red Hat Enterprise Linux 7 HPC: rhel-7-hpc-node-dotnet-rpms
Red Hat Enterprise Linux 8	AMD64 and Intel 64 (**x86_64**) IBM Z and LinuxONE (**s390x**) 64-bit Arm (**aarch64**)	dotnet-sdk-6.0	Appstream NOTE: The AppStream repositories are enabled by default in Red Hat Enterprise Linux 8.
Red Hat Enterprise Atomic Host	AMD64 and Intel 64 (**x86_64**)		
OpenShift Container Pla tform 3.11 and later	AMD64 and Intel 64 (**x86_64**)		
OpenShift Container Pla tform 4.2 and later	IBM Z and LinuxONE (**s390x**)		

CHAPTER 5. CUSTOMER PRIVACY

Various Microsoft products have a feature that reports usage statistics, analytics, and various other metrics to Microsoft over the network. Microsoft calls this Telemetry. Red Hat is disabling telemetry because we do not recommend sending customer data to anyone without explicit permission.

CHAPTER 6. SUPPORT

Red Hat and Microsoft are committed to providing excellent support for .NET and are working together to resolve any problems that occur on Red Hat supported platforms. At a high level, Red Hat supports the installation, configuration, and running of the .NET component in Red Hat Enterprise Linux (RHEL). Red Hat can also provide "commercially reasonable" support for issues we can help with, for instance, NuGet access problems, permissions issues, firewalls, and application questions. If the issue is a defect or vulnerability in .NET, we actively work with Microsoft to resolve it.

.NET 6.0 is supported on RHEL 7, RHEL 8, RHEL 9, and Red Hat OpenShift Container Platform versions 3.3 and later.

See .NET Core Life Cycle for information about the .NET support policy

6.1. CONTACT OPTIONS

There are a couple of ways you can get support, depending on how you are using .NET.

- If you are using .NET on-premises, you can contact either Red Hat Support or Microsoft directly.
- If you are using .NET in Microsoft Azure, you can contact either Red Hat Support or Azure Support to receive Integrated Support.

Integrated Support is a collaborative support agreement between Red Hat and Microsoft. Customers using Red Hat products in Microsoft Azure are mutual customers, so both companies are united to provide the best troubleshooting and support experience possible.

- If you are using .NET on IBM Z and LinuxONE, you can contact Red Hat Support. If the Red Hat Support Engineer assigned to your case needs assistance from IBM, the Red Hat Support Engineer will collaborate with IBM directly without any action required from you.

6.2. FREQUENTLY ASKED QUESTIONS

Here are four of the most common support questions for Integrated Support.

1. When do I access Integrated Support?
   You can engage Red Hat Support directly. If the Red Hat Support Engineer assigned to your case needs assistance from Microsoft, the Red Hat Support Engineer will collaborate with Microsoft directly without any action required from you. Likewise on the Microsoft side, they have a process for directly collaborating with Red Hat Support Engineers.

2. What happens after I file a support case?
   Once the Red Hat support case has been created, a Red Hat Support Engineer will be assigned to the case and begin collaborating on the issue with you and your Microsoft Support Engineer. You should expect a response to the issue based on Red Hat’s Production Support Service Level Agreement.

3. What if I need further assistance?
   Contact Red Hat Support for assistance in creating your case or with any questions related to this process. You can view any of your open cases here.

4. How do I engage Microsoft for support for an Azure platform issue?
   If you have support from Microsoft, you can open a case using whatever process you typically would follow. If you do not have support with Microsoft, you can always get support from Microsoft Support.

6.3. ADDITIONAL SUPPORT RESOURCES

The Resources page at Red Hat Developers provides a wealth of information, including:

- Getting started documents
- Knowledgebase articles and solutions
- Blog posts

.NET documentation is hosted on a Microsoft website. Here are some additional topics to explore:

- .NET
- ASP.NET Core
- C#
- F#
- Visual Basic

You can also see more support policy information at Red Hat and Microsoft Azure Certified Cloud & Service Provider Support Policies.

CHAPTER 7. KNOWN ISSUES

The known issues for running .NET on Red Hat Enterprise Linux (RHEL) include the following:

it does not run on earlier versions of RHEL.

1. **dotnet dev-certs https --trust** does not work on RHEL.
   .NET supports the creation of HTTPS certificate through **dotnet dev-certs https**, but it does not support trusting them through **dotnet dev-certs https --trust**. The client that connects to the ASP.NET Core application, such as **curl** or Firefox, will warn about the untrusted self-signed certificate. To work around this in a browser such as Firefox, ignore the warning and trust the certificate explicitly when the warning about the untrusted certificate comes up. Command-line tools support flags to ignore untrusted certificates. For **curl**, use the **--insecure** flag. For **wget**, use the **--no-check-certificate** flag.

2. There are no NuGet packages for **s390x** on nuget.org.
   Using the **rhel.8-s390x** or **linux-s390x** runtime identifier can cause some **dotnet** commands to fail when they try to obtain these packages. These commands are either not fully supported on **s390x** as described in the other known issues, or the issue can be fixed by not specifying the runtime identifier.

3. Single file applications are not supported on **s390x**.

4. PublishReadyToRun/crossgen is not supported on **s390x**.

5. .NET 6.0 on **s390x** does not understand memory and cpu limits in containers.
   In such environments, it is possible that .NET 6.0 will try to use more memory than allocated to the container, causing the container to get killed or restarted in OpenShift Container Platform. As a workaround you can manually specify a heap limit through an environment variable: **MONO_GC_PARAMS=max-heap-size=<limit>**. You should set the limit to 75% of the memory allocated to the container. For example, if the container memory limit is 300MB, set **MONO_GC_PARAMS=max-heap-size=225M**.

6. The default version of the **Microsoft.NET.Test.Sdk** package in the test project templates (**xunit**, **nunit**, **mstest**) is unusable on **s390x**. Trying to build/run tests will fail with a "System.NotSupportedException: Specified method is not supported" exception.
   If you are trying to run test on **s390x**, update the version of the **Microsoft.NET.Test.Sdk** package to at least 17.0.0.

7. OmniSharp, the language server used by IDEs like Visual Studio Code, is not available on **s390x**.

8. RHEL 9 has disabled several weak security algorithms to improve security.
   Some .NET APIs using these algorithms will fail at runtime with CryptographicExceptions. If you really must use the weak algorithms and risk compromising security, you can loosen the system's security policies by using:

   ```
   # update-crypto-policies --set DEFAULT:SHA1
   ```

   or

   ```
   # update-crypto-policies --set LEGACY”
   ```

   For more information, see the “Security” section in the overview of major changes in the RHEL 9 release notes.

9. Strong Naming will not work out of the box on RHEL 9.
   RHEL 9 has disabled the use of SHA-1 in the default configuration. .NET uses SHA-1+RSA to identify assemblies that have been signed with a strong name. The explicit SHA-1+RSA algorithm combination is a part of the ECMA-335 specification involving strong naming. However, given the recent attacks against SHA-1, RHEL 9 has deprecated the use of SHA-1 (when combined with RSA) to improve security across the entire operating system. This means that any use of strong naming, including verification at build time, will fail.

   The OpenSSL errors on RHEL 9 will indicate an invalid digest algorithm. For example:

   ```
   error : Unhandled exception. Interop+Crypto+OpenSslCryptographicException:
   error:03000098:digital envelope routines::invalid digest
   ```

   There are several possible workarounds:

   - Enable support for SHA-1+RSA, by loosening the system's security policies:

     ```
     # update-crypto-policies --set DEFAULT:SHA1
     ```

     NOTE

     This will not work when FIPS is enabled. In FIPS mode, SHA-1 is completely disallowed.

   - Switch to Public Signing. In order to do this, you must modify the project files to set up a number of properties:

     ```
     <PropertyGroup>

     <AssemblyOriginatorKeyFil>$(MSBuildThisFileDirectory)Key.snk</AssemblyOriginatorKeyF
     le>
       <SignAssembly>true</SignAssembly>
       <PublicSign Condition="'$(OS)' != 'Windows_NT'">true</PublicSign>
     </PropertyGroup>
     ```

10. The NTLM technology is considered insecure in RHEL 9.
    The **gss-ntlmssp** package, which provides NTLM authentication support, has been removed from RHEL 9. That means .NET in RHEL 9 can not authenticate against NTLM. If you use NLTM authentication, please use another mechanism to authenticate.

    For more details, see the Identity Management section of Considerations in Adopting RHEL 9 .