Micro Focus
VisiBroker 8.5 SP8

Release Notes

Contents

Micro Focus VisiBroker 8.5.8 Release Notes

Installing VisiBroker

Before Installing SP8

This release updates VisiBroker 8.5. Before installing this Service Pack you must have VisiBroker 8.5 installed. If you do not have VisiBroker 8.5, you should download it and follow the instructions in the *Installation Guide*.

Installing SP8

To install this release on top of an existing VisiBroker 8.5 installation:

1. Download the release archive to your `VBROKERDIR` folder.

2. Unpack the archive in the same folder.

3. Restart the application.

Upgrading an earlier installation to JDK 11 or 17

VisiBroker 8.5 SP8 supports Java 11 and 17. VisiBroker product installers for 8.5 SP5 and older versions could only be used with a Java JDK 8 or below. If your VisiBroker product installation was performed with an installer for 8.5 SP5 or an older version, and you want to migrate your installation to JDK 11 or higher, there are some additional steps required after the installation described above. This process is not necessary if you are upgrading from 8.5 SP6 or SP7.

Note: Micro Focus strongly recommends that you create backups of all files before making any modifications.

1) Locate all the VisiBroker `.config` files in use by your installation and your applications. The default file names and their locations are:

```
[VBROKERDIR]/bin/toolsjdk.config
[VBROKERDIR]/bin/vbconsole.config
[VBROKERDIR]/bin/vbj.config
[VBROKERDIR]/bin/vbjc.config
```

If you have modified or extended your application's VisiBroker configuration, you might also have additional files. Check your start-up scripts for the argument –`VBJconfig`.

Note: `.config` files can also include further `.config` files with the `'include'` directive.

2) For each `.config` file:

a) Locate and update any `'javahome'` directive lines to point to your new JDK 11 location. For example, replace:

```
javahome /usr/lib64/jvm/java-1.8.0
```

with:

```
javahome /usr/lib64/jvm/java-11
```

b) Remove any lines that specify the VM property `'vmprop java.endorsed.dirs'`. For example, remove:

```
vmprop java.endorsed.dirs=$var(installRoot)/lib/endorsed
```

This JVM option is no longer supported by JDK 11 and higher and generates an error if not removed.

c) Remove any directives that reference `.jar` files that are no longer present in the JDK. For example, remove all entries like:

```
addpath $var(JDK_HOME)/lib/tools.jar addpath
$var(JDK_HOME)/jre/lib/rt.jar
```

d) To each `.config` file, add entries like those below. These specify the location of CORBA classes that have been removed from the JDK:

```
addendorsejar $var(installRoot)/lib/endorsed/vbjendorse.jar addomgjar
$var(installRoot)/lib/visi-omg.jar
```

If you do not make these additions, you might see `ClassNotFoundExceptions` for classes in the `org.omg.*` packages.

e) Remove any entries using the directives "`addbootjars`" or "`addbootpath`". Such entries do not normally occur in default VisiBroker configurations, but you might have added them. As their corresponding JVM options `-Xbootclasspath` and `-Xbootclasspath/p` are no longer supported by JDK 11, using either of these two directives causes a JVM Loader error.

3) If the JDK 11 you want to use is 64-bit, and VisiBroker was installed with a 32-bit JDK version, then apply the appropriate **VisiBroker 64-bit Service Overlay** so that Java services and tools will work with the new JDK. To accomplish this, run the 32-bit VisiBroker installer as normal, but select your 64-bit JDK selected during the installation process. Immediately after installation, unpack the appropriate **VisiBroker 64-bit Service Overlay** patch into the installation directory. Contact Micro Focus SupportLine (https://supportline.microfocus.com/) for further information.

Platforms and Compilers

For a full list of platforms and compilers supported by VisiBroker 8.5 SP8, see the ***VisiBroker 8.5 SP8 Platform Support Notes*** (https://www.microfocus.com/documentation/visibroker/visibroker858/pdfdoc/Micro_Focus_VisiBroker_Platform_Support_Notes_858.pdf).

Notes:

1. Support for Red Hat Enterprise Linux 9 has been added.
2. Support for Windows Server 2022 has been added.
3. Support for Visual Studio 2022 has been added. The version of VisiBroker for Windows (patch filename `08.05.00.p8_opt_vs2015_x64.ojdk8.zip`, product installer filename `vb8508.05.00.P8-winvs2015-x64.exe`) should be used for any applications built with Visual Studio 2015, 2017, 2019, and 2022.

Features

VisiBroker 8.5 SP8 provides enhancements in the following areas.

OpenSSL

The version of OpenSSL used in VisiBroker 8.5 SP7 was susceptible to a Denial-of-Service risk. This issue is described in the OpenSSL security advisory CVE-2022-0778. The OpenSSL version has been updated to v1.1.1n and the issue no longer exists.

SC:02055404

Name Service password-less authentication to a JDBC database

The property `vbroker.naming.loginPwd` cannot be left blank. This is an error condition. To establish a password-less JDBC connection a new property: `vbroker.naming.noLoginCredentials` has been added. Set `vbroker.naming.noLoginCredentials=true` to use this capability, with the default being `false`. Users should then omit the property `vbroker.naming.loginPwd`.

SC:00418836

Secure corbaloc Interoperable Object References

Previously only IIOP protocol forms of corbaloc IORs were supported by the Java and C++ ORBs. Following this enhancement, the ORB will now accept secure corbaloc IORs for the protocol SSLIOP. See the new section on Secure Object URIs in the *VisiBroker Security Guide* for details.

SC:00365610

Java Utility to Replace the Host String in Existing IOR with a New Host String

00365610 (361340)

VisiBroker 8.5 SP8 includes a Java utility that enables you to replace the host string in your existing IOR with a new host string. You can specify a specific host to be replaced or you can choose to replace all hosts in the IOR with the new host.

Usage:

```
vbj com.microfocus.orb.PatchIOR -f [old_host] -r new_host IOR
```

where:

`old_host`	specifies the host to be replaced in IOR. If not specified, all hosts in IOR will be replaced with the new host value.
`new_host`	specifies the URL for the IOR.
`IOR`	specifies the URL for the IOR.

Alternatively, you can use the following format of the command to display usage information and exit:

```
vbj com.microfocus.orb.PatchIOR <-h|-help|-usage|-?>
```

Examples:

```
vbj com.microfocus.orb.PatchIOR -f 127.0.0.1 -r new.host bank.ior

vbj com.microfocus.orb.PatchIOR -f -r new.host file:bank.ior

vbj com.microfocus.orb.PatchIOR -f -r new.host IOR:000000000000001c49...
```

User Documentation

Documentation released with this Service Pack is available online, from
https://www.microfocus.com/documentation/visibroker/visibroker858/.

Java Version Support

- VisiBroker 8.5 SP8 has been rebuilt using Java 8 and so the Java ORB no longer supports Java 7 (or lower) versions of the JDK.
- Support for Java 11 has been added to HPUX.
- Support for Java 11 has been added to Solaris 11.
- Support for Java 17 has been added to the Linux and Windows platforms.
  SC:02190245

Known Issues

There are no known issues with this release.

Resolved Issues

The resolved issues that customers have reported are listed in this section. The numbers that follow each issue are the Reported Problem Incident number followed by the Customer Incident Numbers (in parentheses). RPIs that have numbers only (and no text) are included to confirm that the RPIs have been fixed, since no further information is required.

- Previously running VisiBroker applications with Java 11 could emit a warning like:

  ```
  WARNING: An illegal reflective access operation has occurred
  ```

  This issue has now been fixed and the warning no longer occurs.

  (RPI 636558, RPI 636559)

- Compiling generated code from IDL with Java 11 could result in warnings regarding the use of deprecated constructors for the following classes:
  ```
  java.lang.Boolean
  java.lang.Integer
  java.lang.Long
  java.lang.Short
  java.lang.Float
  java.lang.Double
  ```

```
java.lang.Character
java.lang.Byte
```

This issue has been fixed and no longer occurs. IDL files must be reprocessed with idl2java to observe this fix take effect.

(RPI 636582)

- The use of TLSv1.3 was not supported when using Java NIOSSL in SP7. When using NIOSSL, explicitly configuring the use of TLSv1.3 could fail at the TLS handshake. This issue has now been fixed and TLS v1.3 is now the default maximum level for NIOSSL connections.

    (RPI 647741)

- The Micro Focus® | CORBA® Add-on for Cloud, Containers & Virtual Environments enables you to extend your CORBA applications to operate with Micro Focus CORBA products, including VisiBroker 8.5.8, in the Cloud, in virtual environments, and in container-based platforms such as Docker.

    Some container products do not permit UDP communications across container boundaries, which would prevent the VisiBroker Smart Agent (osagent) from operating. VisiBroker now includes the Smart Agent Relay (osarelay), which enables the osagent to operate across the container boundary. It is available in product distributions for Linux and Windows.

    For further information about the Smart Agent, see the VisiBroker (for C++ or Java) **Developer's Guide** chapter entitled *Using the Smart Agent*. For further information about the Smart Agent Relay, see the ***Micro Focus® | CORBA® Add-on for Cloud, Containers & Virtual Environments Installation and Configuration Guide***. RPI 640948

- The VisiBroker Java ORB will no longer remarshal the VisiBroker ObjectKeys it reads from IORs By default. Previously, if it read a VisiBroker Object Key that was written in 'Little Endian' form, it would switch it to 'Big Endian' (the Java endianness format) before using it. This would happen, for example, when making an invocation onto a C++ VisiBroker server. This could cause issues traversing CORBA-aware firewalls. The Java ORB now always retains the ObjectKey verbatim in the form in which it was originally written in the IOR. To switch back to the previous version's behaviour, set the new property `"vbroker.orb.remarshalObjectKeys"` to `"true"`. RPI 640947

- This change to the VisiBroker IDL compiler enables it to accept IDL-RS annotations and silently ignore them. This change ensures that the IDL compiler can continue to compile IDL files as if they did not contain any IDL-RS annotations. RPI 641830

- Some libraries and executables newly had no version stamping at SP7 (readable with the utility vbver). The affected files were:

```
vaultgen_cpp/vaultgen_cpp.exe
libvbsec.so/libvbsec64.so/vbsec.dll/vbsec64.dll
asrt.jar
vbdev.jar
vbjdev.jar
xmlrt.jar
```

For these files the output of vbver looked like:

```
/opt/VB_Install_64bits/bin/vbver: No version information available
for asrt.jar
```
This problem has been fixed and the files are stamped correctly.
Defect 137012

- A bug in VisiBroker C++ ORB meant that repeated calls to `ORB_init()` followed by `orb->destroy()` would leak memory. This has been fixed and memory use growth no longer occurs in this circumstance.
  RPI584159

- Previously, VisiBroker Java included Log4j 1.2.8 and used it for logging functions within all VisiBroker java processes. The 1.x series of Log4j is no longer maintained by the project owners and has some identified security vulnerabilities. The version of Log4j in the VisiBroker product has been updated to 2.18.0 which is not vulnerable. Log4j 1.2.8 was delivered within VisiBroker as a single jar file: `log4j.jar`. This file is still present in the VisiBroker distribution `./lib` directory. However, its only content following the update is a manifest with `classpath` entries that refer to the Log4j2 components and their dependency jars. These are:

  ```
  jackson-annotations-2.13.3.jar
  jackson-core-2.13.3.jar
  jackson-databind-2.13.3.jar
  jackson-dataformat-xml-2.13.3.jar
  log4j-1.2-api-2.18.0.jar
  log4j-api-2.18.0.jar
  log4j-core-2.18.0.jar
  org.osgi.core-6.0.0.jar
  stax2-api-4.2.1.jar
  woodstox-core-6.2.8.jar
  ```

  Existing scripts and tools that refer only to log4j.jar should not need modification. However, users that re-package VisiBroker into their application should make sure all these new files are included into the same location as `log4j.jar`.

  Users should also note that the Log4j 2.x default log level is now ERROR. Previously, in log4j 1.x, it was DEBUG so users that do not currently explicitly set values, for example `vbroker.orb.debugLevel`, will find they will see less information than previously. See the *Java Developers Guide* for information on properties to control logging.

  The current version of Log4j2 requires a minimum Java 8 JRE version. All VisiBroker 8.5 SP8 jars have been compiled to a Java 8 bytecode version to match this.
  SC:02050498

- In certain virtualised hosting scenarios an ABORT could be generated in function `getcorenum` in the VisiBroker C++ library: `libcpuid(64).so` on Linux platforms. This library has been removed from the product, so the problem can no longer occur.
  SC:02133970

- There were invalid comments in the ORB IDL files that prevented generated Java code from being processed by the Java 8 Javadoc tool. These have been fixed so that the source generated can now be successfully processed by Java 8 Javadoc.
  Defect 254009

- The java2iiop and java2idl tools failed with Java 17 with the error: "`fatal error: internal problem: unexpected exception:`

`java.lang.reflect.InaccessibleObjectException`". The problem has been fixed and this error no longer occurs.
Defect 260091

- When VisiSecure was configured to use both secured and unsecured communication for the same POA, `CORBA::INV_POLICY`, `CORBA::TIMEOUT` and `'int'` exceptions could be seen to be thrown and caught internally to the implementation under a debugger. These hampered user application debugging. These exceptions are no longer used in the implementation in these circumstance, so they are no longer visible in users' debugging sessions.
SC:02326525

- From VisiIbroker 8.5 SP4 onwards, when using VisiSecure C++, a bug prevented correct use of ClientContextID to correctly identify subsequent messages within an established security context. This could cause a `NO_PERMISSION` exception to be created in the server in some circumstances which then caused a `REMARSHAL` exception in the secure client application and retransmission of the client request. This has been fixed and the problem no longer occurs.
Defect 313021

- As of Java 17, both the `SecurityManager` (https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/SecurityManager.html) and the `RMISecurityManager` (https://docs.oracle.com/en/java/javase/17/docs/api/java.rmi/java/rmi/RMISecurityManager.html) have been marked for removal. As per https://openjdk.org/jeps/411, a new warning that cannot be suppressed is emitted to indicate the future removal of these features. This warning is displayed when using the RMI features included in VisiBroker 8.5 SP8 with Java 17. We have now implemented a new property, `vbroker.orb.useRMISecurityManager` which, when set to false, disables the RMISecurityManager used in RMI operations so that no warnings appear with Java 17. This is the default with Java 17 and upwards. This property defaults to `true` for versions of Java less than Java 17. If you want to re-enable the RMISecurityManager for Java 17, you might set `vbroker.orb.useRMISecurityManager=true`. This then causes the original warnings to be seen.
Defect 330006

- When running a VisiBroker Java application with Java 11, with security enabled, with DEBUG level logging set, and using the `vbjclientorb.jar` in place of `vbjorb.jar`, a `java.lang.NoClassDefFoundError: javax/rmi/CORBA/Util` error was encountered. This happens because the package javax.rmi.CORBA was removed from the Java runtime in Java 11. The dependency on `javax.rmi.CORBA.Util` has been made conditional such that an attempt to use this class is only made when the class is present in the runtime environment and the error no longer occurs.
SC:02400777

Removal of VisiBroker GUI components

The following GUI components are no longer automatically installed:

- vbconsole
- JDSExplorer

These are now available as an optional component. To install, please download and extract the GUI components archive and extract it onto your product installation.

The GUI components archive can be downloaded from our Software License & Download (SLD) site, which is part of the Micro Focus Support Portal. Among other benefits, this site provides access to product license keys and install-kits, including the relevant GUI components.

For more information on the SLD, please see our Support Portal overview:

https://support.microfocus.com/help/support-portal-overview.pdf

The GUI components have been separated from the product as they do not fully comply with the Section 508 and WCAG accessibility requirements and guidelines. It is intended that future product releases will include updated GUI components that comply with the relevant accessibility guidelines.

Updates and SupportLine

Our Web site gives up-to-date details of contact numbers and addresses.

Further Information and Product Support

Additional technical information or advice is available from several sources.

The product support pages contain a considerable amount of additional information, such as:

• The WebSync service, where you can download fixes and documentation updates.
• The Knowledge Base, a large collection of product tips and workarounds.
• Examples and Utilities, including demos and additional product documentation.

To connect, enter https://www.microfocus.com in your browser to go to the Micro Focus home page.

Note: Some information may be available only to customers who have maintenance agreements.

If you obtained this product directly from Micro Focus, contact us as described on the Micro Focus Web site, https://www.microfocus.com. If you obtained the product from another source, such as an authorized distributor, contact them for help first. If they are unable to help, contact us.

Disclaimer

This software is provided "as is" without warranty of any kind. Micro Focus disclaims all warranties, either express or implied, including the warranties of merchantability and fitness for a particular purpose. In no event shall Micro Focus or its suppliers be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if Micro Focus or its suppliers have been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of liability for consequential or incidental damages so the foregoing limitation may not apply.

Micro Focus is a registered trademark.

Copyright © Micro Focus 2022. All rights reserved.