Contacting Parasoft Technical Support

This topic explains several ways to contact technical support, as well as how to prepare and send "support archives" that help the technical support team diagnose any problems you are experiencing.

In this section:

Obtaining Live Online Support (Windows only)

The Parasoft support team is available online to answer your questions. This live support allows you to chat in real-time with the support team and perform desktop sharing if needed. To receive live online support, go to https://www.parasoft.com/support/ . This live tech support feature currently supports only the Microsoft Windows operating system.

Creating a Support Case

You can file a support case directly at https://parasoft.force.com/customerportal/CommunityNewCasePage.

Contacting us via Phone or E-mail

USA Headquarters

Tel: (888) 305-0041 or (626) 256-3680

Email: support@parasoft.com

Other Locations

See http://www.parasoft.com/contacts.

Preventing C++test from Running Out of Memory

To prevent the product from running out of memory, you can add two memory parameters to the script or shortcut being used to start the product:

- The initial size of the JVM (Xms).
- The maximum size of the JVM (Xmx).

Typically, both are set to the same size (for instance, 256MB). However, if you have occasional problems but don't want to always allocate a large amount of memory, you can set the parameters to different sizes (for example, 256MB as the initial size and 512MB for the maximum size).

Examples:

- C++test standalone: `cpptest.exe -J-Xms1024m -J-Xmx1400m`
- C++test plugin for Eclipse: `eclipse.exe -vmargs -Xmx1400m`

Note that the maximum size you can set depends on your OS and JVM.

If you are running the Eclipse plugin under Sun Java 1.5 and get a `java.lang.OutOfMemoryError: PermGen space` error messages, start Eclipse with `eclipse -vmargs -XX:MaxPermSize=256m`

Limiting the Number of Open Files

The "osgi.bundlefile.limit" property specifies a limit on the number of jar files the Eclipse framework will keep open. In Eclipse 3.5 and later, this property defaults to 100. In Parasoft Test for Windows the value of this property is set to 0 (unlimited). On the other platforms, it is set to 300.

Increasing the value of this property or setting it to 0 (unlimited) has been know to significantly improve performance. However, increasing the value of this property may cause crashes or a "Too many open files" runtime exception.

To return to the Eclipse bundle file limit, start the application with the following parameter:

- standalone: `-J-Dosgi.bundlefile.limit=100`
- plugin: `-vmargs -Dosgi.bundlefile.limit=100`

Preparing a "Support Archive" and Sending it to Technical Support

If you are experiencing testing problems such as build failures, the best way to remedy the problem is to create a zip archive containing the source file(s) that caused that failure (if applicable), as well as related test information, then send that zip file to Parasoft's support team. To facilitate this process, you can have the product automatically create an archive when testing problems occur. On average, these archives are about half a megabyte, and are created in about one minute.
By default, an archive is not created when testing problems occur. You can either manually prepare and send a support archive when needed, or you can modify Parasoft archive creation options so that the product automatically prepares and sends an archive when testing problems occur.
To configure the product to automatically prepare and send archives when testing problems occur:

1. Open the Technical Support panel by choosing **Parasoft> Preferences**, then selecting the **Parasoft> Technical Support** category.
2. Check **Enable auto-creation of support archives**.
3. Customize additional options as needed. Available options are:

Option	Description
Send archives by e-mail	If this option is enabled, the archive is sent automatically. If you enable this option, be sure to set the e-mail options in **Preferences> E-mail** if you have not already done so.
C++test configuration files	All user and team Test Configurations available to C++test at the time the technical support archive is created.
DTP engine configuration files	Additional C++test data generated during analysis.
Environmental data	Environment variables, JVM system properties, platform details, additional properties (memory, other).
Preprocessed source files	If this option is enabled, you will be prompted to include preprocessed source files when the **Create Archive** button is clicked. If the **Enable auto-creation of support archives** option is enabled, pre-processed source files will be automatically added to any archives created.
Project properties	Properties from the project, platform, tool, and options extractor. *If this is enabled, the archive will probably contain your preprocessed sources, Makefiles, and other files you have added to the project. Moreover, it might also contain data that can reveal your project's structure.*
Test run data	Checker-specific (standards, execution, generation) data that is automatically collected during test sessions. This option must be enabled in order to collect data during test runs.
General application logs	Various platform/application logs.

Enable verbose logging	Includes verbose logs included in the archive. Note that this option cannot be enabled if the logging system has custom configurations. • Verbose logs are stored in the `xtest.log` file within the user-home temporary location (on Windows, this is `<drive>:\Documents and Settings\<user>\Local Settings\Temp\parasoft\xtest`). • Verbose logging state is cross-ses-sionpersistent(restored on application startup). • The log file is a rolling file: it won't grow over a certain size, and each time it achieves the maximum size, a backup will be created.
Enable source control output	Includes output from source control commands in verbose logs. Note that the output could include fragments of your source code.
Advanced options	If the support team asked you to enter any advanced options, check this box, then enter the options here.
Archives location	If you do not want to use the listed archive location, specify a new one.

4.Click **Apply**, then **OK**.

To manually create a support archive:

- Choose **Parasoft> Preferences**, select the **Technical Support** category, select the desired archive options, then click **Create Archive**.

To open the Technical Support Archive Manager, which allows you to review, e-mail, or delete recent support archives:

- Choose **Parasoft> Preferences**, select the **Technical Support** category, then click **Browse Recent Archives**.

When creating a support archive it is best to ensure that it contains all the info which is relevant to the problem and does not contain any unrelated info.

Best Practice: Creating an Archive with the Most Relevant Data

When a technical support archive is created, the complete application logs are included. The logs may contain information from many test runs over a long period of time—but chances are that only a small part of that information is relevant to the problem you are experiencing. To help technical support isolate the cause of the problem, create a technical support archive containing application logs for only the testing session that produces problems. To do this:

1. Clean application logs by turning on verbose logging. If verbose logging is already enabled, then disable it and re-enable it.
2. Run the testing session that causes problems.
3. Prepare a technical support archive.

Adjusting Timeouts

If technical support suspects that your problems stem from network or connection problems, they may advise you to adjust timeouts by adding the following switches to your JVM:

Switch	Description	Example
parasoft.ws.timeout	Sets socket timeout for all services. Value is in seconds	`-Dparasoft.ws.timeout=60`
parasoft.ws.connection.timeout	Sets connection timeout for all services. Value is in seconds.	`-Dparasoft.ws.connection.timeout=15`
parasoft.tcm.timeout	Sets timeout for Team Server service. It could set timeout only for this service or override the value set by the general parasoft.ws.timeout, value in seconds.	`-Dparasoft.tcm.timeout=30`

If only parasoft.ws.timeout is set, it affects both socket and connection timeout.