



STINGRAY STUDIO 2019 RELEASE NOTES

Stingray® Studio



Introduction

In this Document

These release notes contain a summary of new features and enhancements, late-breaking product issues, migration from earlier releases, and bug fixes.

For change logs or other information on previous releases, please contact Technical Support.

NOTE >> Rogue Wave products may contain undocumented interfaces. These interfaces are not supported for general use and may be changed or removed from release to release.

NOTE >> The version of this document in the product distribution is a snapshot at the time the product distribution was created. Additional information may be added after that time because of issues found during distribution testing or after the product is released. To be sure you have the most up-to-date information, see the version of this document on the Rogue Wave web site:

<https://docs.roguewave.com/en/stingray/current/>



New Features

Updated Platform Support

Please see Platforms and System Requirements at <https://docs.roguewave.com/en/stingray/current/>.

In this version, support was added for current updates of **Windows 10** and **Visual Studio 2017 and 2019**, in 32- or 64-bits.

Support for Visual Studio 2013 was dropped for this release.

For both Visual Studio 2017 and 2019, support was added for C++ language standard (ISO c++14, c++**17** and c++**latest**) and existing code was improved for conformance, to allow building with the `/permissive-` compiler option.

Testing was done on Windows up to version 10.0.17763, Visual Studio up to version 2019 (16.1.5) and .NET Framework up to version 4.7.2

Stingray Assistant ported on Visual Studio 2017 and 2019

Stingray Assistant was ported on Visual Studio 2017 and 2019, the same executable is used on these versions as well as on Visual Studio 2015.

For Visual Studio 2017 and 2019, Assistant was upgraded to allow the following to be set in C++ projects (including Stingray makefile-based):

- Windows SDK version
- C++ language standard
- Additional compiler options (`/permissive-` and others)

These settings are applied to the project when the corresponding solution is launched through the Assistant.



Bug Fixes

Common

- [SRSTUDIO-8103]: Fixed warning LNK4281 using default /DYNAMICBASE linker option
- [SRSTUDIO-8234]: Fixed problem in InstallShield Silent Install
- [SRSTUDIO-8264]: Removed deprecated compiler option /Gm

Grid

- [SRSTUDIO-5711]: Fixed incorrect autodetection of a sort type
- [SRSTUDIO-7462]: Fixed bitmap resizing on a high DPI screen
- [SRSTUDIO-7468]: Fixed checkboxes resizing on a high DPI screen
- [SRSTUDIO-7692]: Fixed CGXDateTimeCtrl resizing on a high DPI screen
- [SRSTUDIO-7626]: Fixed CGXComboBox and CGXComboBoxWnd resizing on a high DPI screen
- [SRSTUDIO-7869]: Fixed CGXTabbedComboBox and CGXTabbedComboBoxWnd resizing on a high DPI screen
- [SRSTUDIO-7654]: Fixed CGXSpinEdit and CGXHotSpotEdit resizing on a high DPI screen
- [SRSTUDIO-8228]: Fixed CGXRadioButton and CGXRadioButtonEx resizing on a high DPI screen
- [SRSTUDIO-8391]: Fixed crash in CGXRichEditCtrl with large text

Toolkit

- [SRSTUDIO-7549]: Fixed size for icons and bitmaps in SECTreeCtrl on a high DPI screen
- [SRSTUDIO-7585]: Fixed wrong size of label and heading text in SECTreeCtrl on a high DPI screen

Views

- [SRSTUDIO-6560]: Fixed hanging of OV applications



Known Issues

Project Files

Targeted to Windows SDK

We ship `.vcxproj` files ported to Visual Studio 2017 targeted to Windows SDK 10.0.17763.0 with following line of code:

```
<WindowsTargetPlatformVersion>10.0.17763.0</WindowsTargetPlatformVersion
```

If you are targeting the Windows 8.1 SDK, this line is not needed and should be removed.

For Visual Studio 2019, we use new option to target to the latest installed version of SDK:

```
<WindowsTargetPlatformVersion>10.0</WindowsTargetPlatformVersion
```

You can re-target C++ projects to another version of the Windows SDK using Windows SDK setting in Stingray Assistant or by manually editing the `.vcxproj` file and change the code line above to match the Windows 10 SDK version you would like to target.

For projects which are not makefile-based, you can re-target the Windows SDK version on the project's property pages (**Configuration Properties | General | Windows SDK Version**) or use the "Retarget Projects" option in Visual Studio IDE. Please, be aware these options do not work for makefile-based projects used for main Stingray components.

Targeted to .NET Framework

Stingray projects which use .NET, are targeted to .NET Framework 4.6.2.

To set this targeting in shipped MFC samples or other similar projects used with C++\CLI, you need to add following code line in the <PropertyGroup Label="Globals"> of the *vcxproj file:

```
<TargetFrameworkVersion>v4.6.2</TargetFrameworkVersion>
```

Libraries Build Random Error

On start of building a solution for Stingray libraries, sometimes an error is displayed in the build log:

```
----- Build started: Project: RWUXTheme, Configuration: Stingray Lib MFC DLL Unicode Release  
(au) x64 -----  
Error: The operation could not be completed. Unspecified error
```

Usually it is enough to close and re-open the solution to overcome this error.



Limitations

Installation

General Installation Issues

NOTE >> Each Stingray Studio release is a replacement for, not an upgrade to, the previous version. Always perform an installation into a new area, and not into the same install directory as a previous version.

- Documentation location changed
User Guides are no longer installed into the Docs folder, but instead are provided online.

These guides are accessible from the *Documentation* shortcut installed in the Start menu or from the Stingray Assistant tool.

Build

C++\CLI Issues

With use of the latest C++ language standards, there are some limitations on build with `/clr` compiler option (C++\CLI). This limitation impacts FoundationEx.NET, Grid for .NET and managed build configuration for all other Stingray components.

- **Visual Studio 2017:** *for each version of C++ language standard, setting the `/clr` option works fine without option `/permissive-`, but build with the `/permissive-` compiler option requires to set an additional option `/Zc:twoPhase-`*
- **Visual Studio 2019:** *build with the `/clr` option is not available for `c++latest` (Command Line error D8016 : '`/clr`' and '`/std:c++latest`' command-line options are incompatible); for other versions of C++ language standard, it works fine without option `/permissive-`, but build with the `/permissive-` requires additional option `/Zc:twoPhase-`*

Stingray Studio Assistant

Missed links

There is no links in Stingray Studio Assistant to navigate to GridExcelHelper and Grid for .NET:

- `<InstallDir>\Src\ GridExcelHelper*.sln`
- `<InstallDir>\Samples\Grid\Excel\ExcelReadWrite_CLI\1stGridS1_*.sln`
- `<InstallDir>\Objective Grid for Microsoft .NET\GridControl*.sln`
- `<InstallDir>\Objective Grid for Microsoft .NET\Samples\Samples*.sln`
- `<InstallDir>\Objective Grid for Microsoft .NET\Tutorials\Tutorials*.sln`

Additional compiler options

There is a new text box which allows you to set the compiler option `/permissive-` for both makefile-based and non-makefile-based projects, it also allows you to set other options for makefile-based projects. For non-makefile-based projects, additional compiler options can be set in Visual Studio, on the project's property pages.

Rogue Wave provides software development tools for mission-critical applications. Our trusted solutions address the growing complexity of building great software and accelerates the value gained from code across the enterprise. Rogue Wave's portfolio of complementary, cross-platform tools helps developers quickly build applications for strategic software initiatives. With Rogue Wave, customers improve software quality and ensure code integrity, while shortening development cycle times. © Rogue Wave Software, Inc. All Rights Reserved