



What's New in CodeSite 5

The **CodeSite Logging System** gives developers deeper insight into how their code is executing, which enables them to locate problems more quickly and ensure their application is running correctly. CodeSite's logging classes let developers capture all kinds of information while their code executes and then send that information to a live display or to a log file. Furthermore, both styles of logging, *live logging* and *file logging*, can be performed locally or remotely.

CodeSite 5 represents the latest major release of the CodeSite Logging System and specifically focuses enhancing and improving several core areas:

- Developer Productivity
- Usability
- Logging Capabilities
- Deployment

The following sections provide specific details on the new features and enhancements included in CodeSite 5:

- Logging Classes
- Live Viewer/File Viewer
- Dispatcher
- Controller
- Method Tracer
- CodeSite Tools

Logging Classes

- CodeSite 5 includes support for **Embarcadero RAD Studio XE2** including 64-bit Delphi support.
- In previous versions of CodeSite, the message type parameter to CodeSite logging methods was used to identify the type of message. However, in certain situations, the *message type* was also used to identify the *message content*. For example, the Object message type was used to indicate that additional properties and values were associated with the message. While this worked well in most cases, the double-duty role of the message type parameter also caused some limitations. For instance, it was not possible to send an object instance to CodeSite and have the message marked as an Error message. The Object message type would override the Error message type.

In CodeSite 5, a message's type is separate from its content. The CodeSite logging classes have been extended to handle the new message content parameter. Also, existing method calls will NOT need to be changed. The new logging methods will select default values for the message type and message content if they are not provided. In addition, both values are visible in the CodeSite Viewers.

- CodeSite 5 utilizes a redesigned message packet when transporting CodeSite messages. In previous versions, the native Delphi logging classes utilized a packet layout combining fixed length fields along with variable length data, while the .NET logging classes utilized a serialized XML layout. In CodeSite 5, all logging classes regardless of platform or framework now utilize the same packet layout. The new layout is more flexible than the fixed length approach and does not have the high overhead of XML.
- The **LogFile**, **TCP**, and **UDP** destination sub-types now define a **Compressed** property. For **LogFile** destinations, setting **Compressed** to True causes the Dispatcher to compress each message before saving it to the log file. This reduces the size of the message on disk and also obfuscates the message data in the log file. For **TCP** and **UDP** destinations, the **Compressed** property instructs the *local* Dispatcher to compress each message before transporting the message to the *remote* Dispatcher.
- The **ConnectUsing*** methods (in **CodeSiteManager** and **TCodeSiteLogger**) in the native Delphi/C++Builder logging classes have been updated to accept a new **MsgFormat** parameter of type **TCodeSiteMessageFormat**. The default parameter value is **mfDefault**. The other value of **TCodeSiteMessageFormat** is **mfCompressed**. When **mfCompressed** is specified, the messages are compressed before they are transported to the Dispatcher. Use the **mfCompressed** message format along with the new **Compressed** options for destinations (see above) and the message data will be compressed and obfuscated throughout the entire transport process.
- **TraceMethod** functionality is now available directly in the **TCodeSiteLogger** class. The implementation is based on the classes that were originally part of the *CodeSiteEx* project, an add-on to CodeSite, created by *Dustin Campbell* and then enhanced and maintained by *Rich*

Ackerson. **TraceMethod** allows both an **EnterMethod** and **ExitMethod** message to be sent with just a single statement. When **TraceMethod** is called, an **EnterMethod** message is sent. When control exits from the current method whether by normal control flow or by exception, an **ExitMethod** message is automatically sent. In addition, the **TraceMethod** method can accept an optional parameter, **tmoTiming**, which can be used to include timing information for the method being traced.

- CodeSite 5 also defines a new **csmTiming** message type, which is used by the new timing capabilities available in the **TCodeSiteLogger** class to log durations. In addition to the **tmoTiming** parameter defined above for **TraceMethod**, **TCodeSiteLogger** also defines a new **Timer** method, which returns an interface reference to a high resolution timer that can be used to time sections of your code.
- All **Send** methods that accept format strings have been enhanced such that if a format exception is raised, the original format string is also included in the error message.
- Bitmap images (e.g. screen shots) are now transported and stored as PNG images instead of standard bitmap images. This results in a significant savings in the size of the log files and transfer packets.
- Added new **Send** method overload that supports sending **TPngImage** images directly. The **Send** method that accepts a bitmap image is still available. The bitmap is converted to a PNG image and then transported.
- The **TCodeSiteLogger.SendVariant** method (Delphi) has been modified to correctly handle additional variant types that have been added to more recent versions of Delphi (e.g. **varUString**, **varLongWord**).
- Fixed issue in evaluating a published variant property that was null.
- Enhanced **SendDateTime** and **WriteDateTime** to accept a format parameter which will override the default **DateTimeFormat** property.
- For each logger instance, the current *Checkpoint* value is now accessible via the **CheckpointCounter** property.
- Use the new **SendFileVersionInfo** method to send version information for the specified file name.
- The **SendSystemInfo** method has been enhanced to record the number of monitors, the primary screen size, and the virtual screen size.
- The **SendControls** method now handles unnamed controls.

- As a result of the new transport message packet, Unicode character encoding issues in .NET have been eliminated (e.g. extended Polish characters).
- The VCL.NET logging classes have been discontinued and removed from the product. The VCL.NET framework has not been supported by Embarcadero for quite some time and the Raize.CodeSiteLogging.dll assembly can be used in .NET and VCL.NET applications. As a result, the Raize.CodeSiteLogging.Vcl.dll assembly is no longer necessary.
- The Raize.CodeSiteLogging.dll assembly for .NET Framework 1.1 has been removed and is no longer supported.
- The packages that contain the CodeSite Logging classes for Delphi/C++Builder development have been renamed slightly to remove the implication that they are tied to the VCL. The CodeSite Logging Classes can be used in any Delphi/C++Builder Windows (32-bit and 64-bit) application including VCL and FMX.

The new runtime package changes are:

CodeSite Studio

CodeSiteLoggingVcl120.bpl ==> CodeSiteLoggingPkg120.bpl

CodeSiteLoggingVcl140.bpl ==> CodeSiteLoggingPkg140.bpl

CodeSiteLoggingVcl150.bpl ==> CodeSiteLoggingPkg150.bpl

CodeSite Express

CodeSiteExpressVcl120.bpl ==> CodeSiteExpressPkg120.bpl

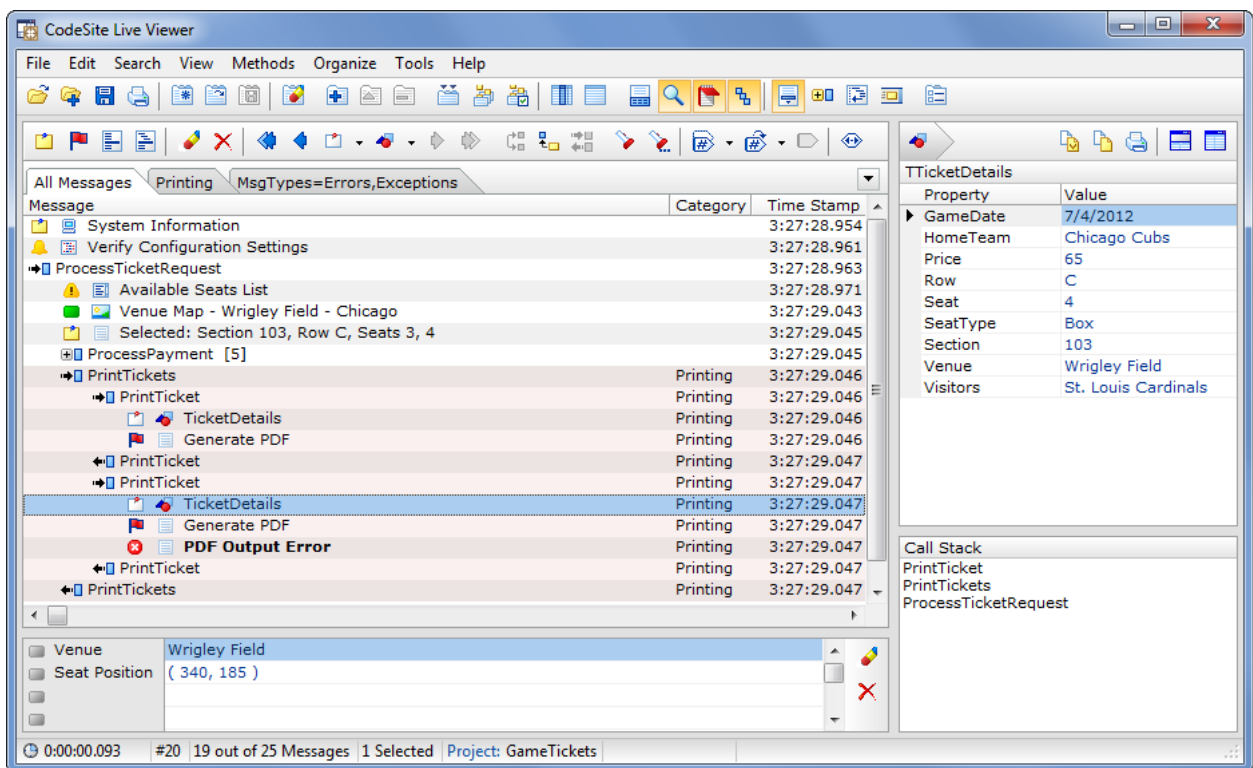
CodeSiteExpressVcl140.bpl ==> CodeSiteExpressPkg140.bpl

CodeSiteExpressVcl150.bpl ==> CodeSiteExpressPkg150.bpl

Note that in your custom packages that require the CodeSite runtime package, you will need to change the requires clause to specify **CodeSiteLoggingPkg** or **CodeSiteExpressPkg** as necessary.

Live Viewer / File Viewer

- As noted in the changes to the logging classes, a message's type has been separated from the message's content. This separation of message type and content is prevalent throughout both viewers. For instance, the message list has been enhanced to display both the *Message Type* and *Message Content* icons for each message. Second, the message navigation toolbar now has drop-downs for both Message Type and Message Content. The Automatic Navigation Mode has also been enhanced to automatically update the type and content values from the selected message. The New View dialog box has also been updated to allow the selection of message types and content values for filtering.
- The viewers have been updated to support loading and saving the new CodeSite 5 log file format. The viewers can also load log files in the CodeSite 3 & 4 log file format.
- The tabs for additional filtered views have been moved to the top of the message list rather than the bottom. The change makes it much easier to manage the views because the commands that affect the views are located at the top of the message list in the message list toolbar.



- The *Ctrl+Tab* and *Ctrl+Shift+Tab* keyboard commands can now be used to navigate between view tabs.
- All view tabs except the All Messages tab now have a close button displayed on the tab to quickly close views that are no longer needed.

- The order of the view tabs can be quickly changed by simply dragging a tab to the desired position. All tabs can be moved except the primary All Messages tab, which remains in the left most position.
- The File Viewer now includes a **Refresh** button on the toolbar.
- The primary font used in the viewer matches the default font used by the operating system. For example, if Windows Vista or Windows 7 is being used, then the font used by the viewers is Seque UI. Earlier versions of Windows use Tahoma.
- The General page of the Options dialog box now contains a color picker to specify the **Highlight Color** used throughout the viewer.
- The Xml Structure and Syntax inspector has been integrated directly into the viewer and, as a result, it is no longer necessary to distribute a separate plug-in package to utilize this inspector.
- The filter criteria used to define new views has been extended to support *Regular Expressions* in the message text field.
- *Regular Expressions* can now be used in the Find dialog box.
- The **Update Synchronized Message** command has been added to the Search menu. Use this command to quickly change the message that is synchronized between multiple views. This command can also be invoked by clicking the desired message while holding down the Alt key.
- Resolved issue where hint windows would not appear correctly if the viewer was not on the primary monitor in a multi-monitor system.
- The viewers are built using Delphi XE2, and as a result, any custom plug-ins will need to be recompiled using XE2 before they can be loaded into the viewers.

Dispatcher

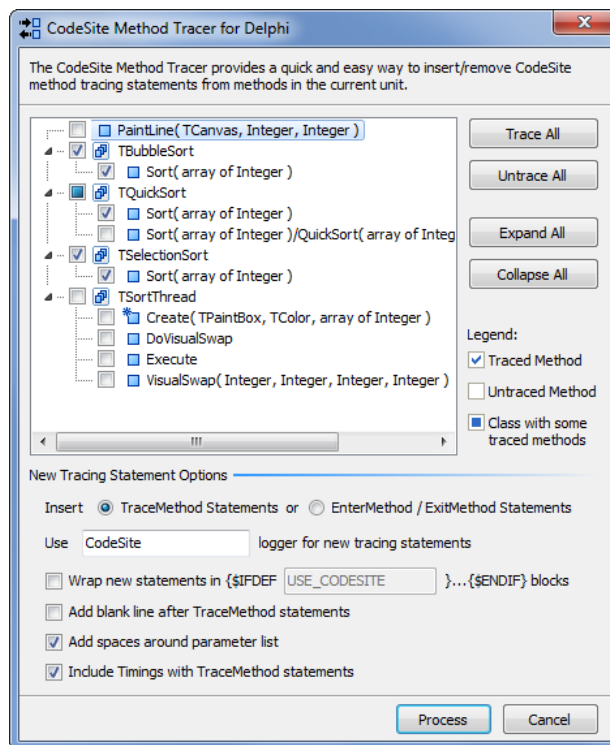
- CodeSite 5 utilizes a redesigned message packet when transporting CodeSite messages. In previous versions, the native Delphi logging classes utilized a packet layout combining fixed length fields along with variable length data, while the .NET logging classes utilized a serialized XML layout. In CodeSite 5, all logging classes regardless of platform or framework now utilize the same packet layout. The new layout is more flexible than the fixed length approach and does not have the high overhead of XML. This change is internal and will have no external effect on your logging code.
- The log file format has changed in CodeSite 5. The new format is more flexible than previous versions and provides support for future enhancements.
- The Monitor Ports option in the Dispatcher Settings has been removed. The CodeSite Dispatcher is now configured to always monitor the TCP and UDP specified ports.
- The Blocked Messages capability has been removed from the CodeSite Dispatcher. Although rarely used, this feature affected dispatching performance.
- The CodeSite 5 Dispatcher will accept CS3 and CS4 messages and automatically convert them into the CodeSite 5 format.
- The CodeSite 2 message format is no longer supported. That is, the CodeSite 5 Dispatcher will not recognize CodeSite 2 messages. CodeSite 2 is very old and applications built with CodeSite 2 will need to be upgraded to work with the CodeSite Tools in CodeSite 5.
- The CodeSite Dispatcher now checks for the presence of the CodeSite Live Viewer at start up. If the Live Viewer is not present in the default installation directory or in the current Dispatcher directory, then a warning message is added to the Dispatcher log. In previous versions, it was possible to flood the Dispatcher log with error messages indicating that the Viewer cannot be started, which can happen if a CodeSite Logger sends messages to simultaneous destinations (including the Viewer) and the Viewer cannot be started.
- Fixed an issue in Dispatcher startup code that under certain circumstances could result in the Dispatcher not responding to messages.
- The Dispatcher now correctly supports creating more than 100 parts in a Log File destination.

Controller

- The Monitor Ports option in the Settings has been removed. The CodeSite Dispatcher is now configured to always monitor the TCP and UDP specified ports.
- The Blocked Messages capability has been removed from the CodeSite Dispatcher. Although rarely used, this feature affected dispatching performance.

Method Tracer

- The Method Tracer for Delphi has been enhanced to support the new **TraceMethod** functionality in CodeSite 5. The tracer supports inserting either **TraceMethod** statements or **EnterMethod/ExitMethod** statements. When **TraceMethod** statements are used, the Method Tracer also supports including the new **tmoTiming** parameter which instructs the logger to add a **csmTiming** message to the log before the method exits.



CodeSite Tools

- The executables included in the CodeSite Tools now are digitally signed.
- Code Templates for RAD Studio have been extended with two new templates for the new **TraceMethod** functionality. Use **cstm** to quickly add a **CodeSite.TraceMethod** statement to your code. Use **cstt** to add a **CodeSite.TraceMethod** statement (including the **tmoTiming** option) to your code.