

Silverlight support for Csharp and VB.NET - 1.2



The **Silverlight extension** is deprecated and no new development will be actioned. All new features and functionality for the support of Silverlight are now handled instead by the **.NET XAML** extension. Please see the section of documentation entitled [Existing extensions for WPF and Silverlight](#) for more information about the .NET XAML extension and the impacts of moving to it.

- [Description](#)
- [In what situation should you install this extension?](#)
- [What's new?](#)
- [Silverlight framework support](#)
- [Function Point, Quality and Sizing support](#)
- [CAST AIP compatibility](#)
- [Supported DBMS servers](#)
- [Prerequisites](#)
- [Download and installation instructions](#)
 - [CAST Transaction Configuration Center \(TCC\) Entry Points](#)
 - [Manual import action for CAST AIP 8.2.x](#)
- [What results can you expect?](#)
 - [Objects](#)
 - [Links](#)
 - [Limitations](#)



Summary: This document provides basic information about the extension providing Silverlight support for C# and VB.NET

Description

This extension provides support for **Silverlight**. The calculation of Automated Function Points for your .NET analyses will be supplemented through the creation of new objects and links specific to the Silverlight framework that will link back to objects/links produced by the base .NET analyzer.

In what situation should you install this extension?

If your .NET application contains Silverlight source code and you want to view these object types and their links, then you should install this extension.

What's new?

- The extension is now shipped with a set of CAST Transaction Configuration Center **Entry Points**, specifically related to **Silverlight**. Please see [CAST Transaction Configuration Center \(TCC\) Entry Points](#) below for more information about this.
- See also: [Silverlight 1.2 - Bug Fix List](#).

Silverlight framework support

The following Silverlight frameworks are supported by this extension:

Version	Supported
All versions up to 5.0	

Function Point, Quality and Sizing support

This extension provides the following support:

- **Function Points (transactions)**: a green tick indicates that OMG Function Point counting and Transaction Risk Index are supported
- **Quality and Sizing**: a green tick indicates that CAST can measure size and that a minimum set of Quality Rules exist

Function Points (transactions)	Quality and Sizing
	

CAST AIP compatibility

This extension is compatible with:

CAST AIP release	Supported	Technology
8.3.x	✓	C# and VB.NET
8.2.x	✓	C# and VB.NET

Supported DBMS servers

This extension is compatible with the following DBMS servers:

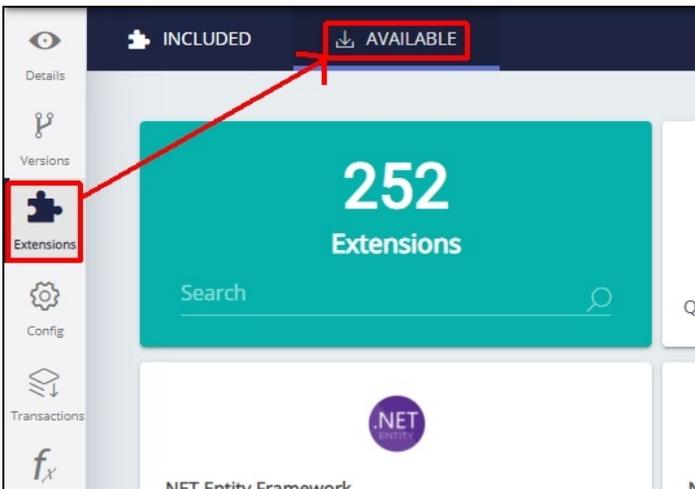
DBMS	Supported
CSS/PostgreSQL	✓
Oracle	✓
Microsoft SQL Server	✗

Prerequisites

- ✓ An installation of any compatible release of CAST AIP (see table above)

Download and installation instructions

Include the extension using the interface in AIP Console:

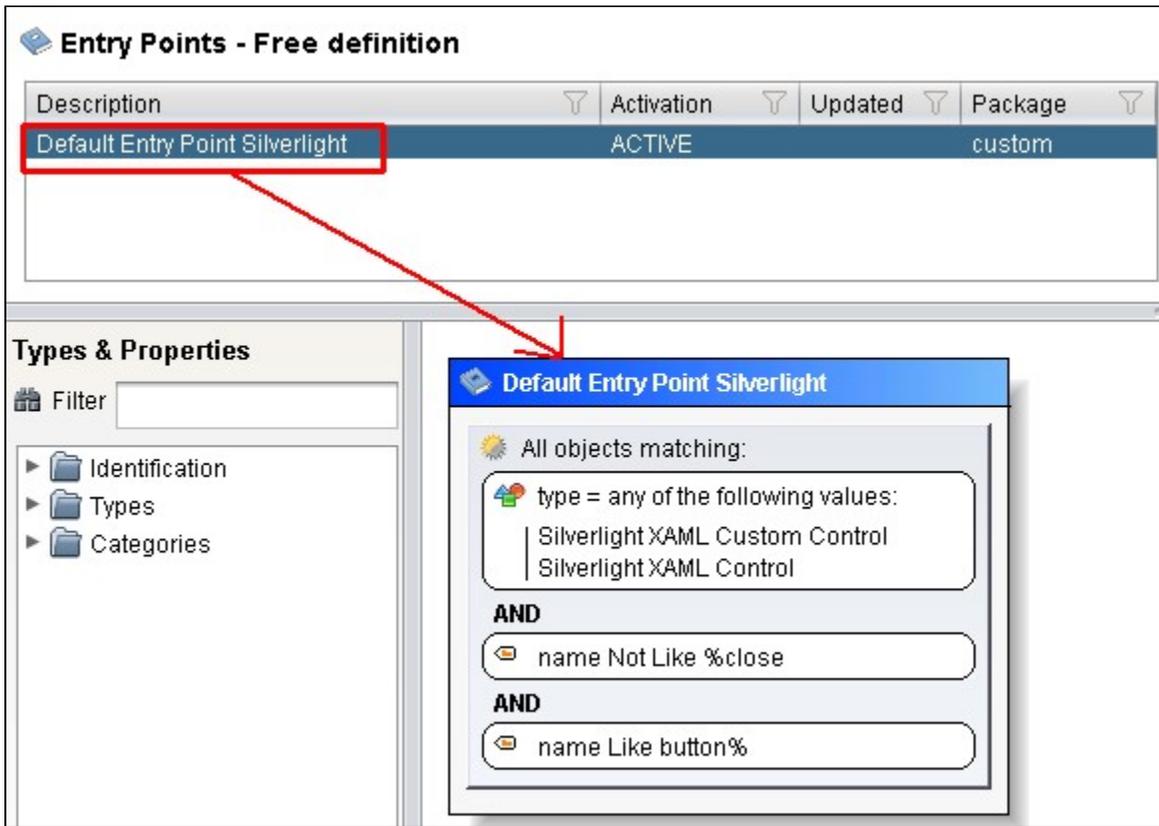


There is nothing further to do. Follow the instructions below to run a new analysis/snapshot to generate new results:

- [Advanced onboarding - run and validate the initial analysis](#)
- [Advanced onboarding - snapshot generation and validation](#)

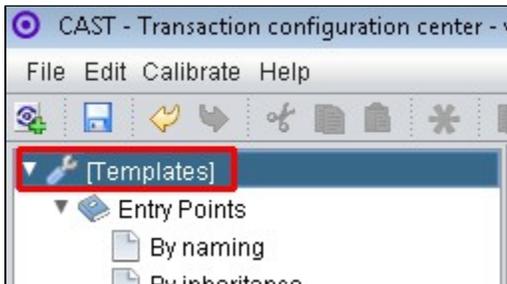
CAST Transaction Configuration Center (TCC) Entry Points

In **Silverlight 1.2.x**, if you are using the extension with **CAST AIP 8.3.x**, a set of Silverlight specific **Transaction Entry Points** are now **automatically imported** when the extension is installed. These **Transaction Entry Points** will be available in the CAST Transaction Configuration Center:

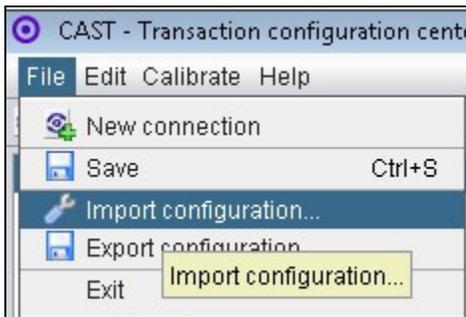


Manual import action for CAST AIP 8.2.x

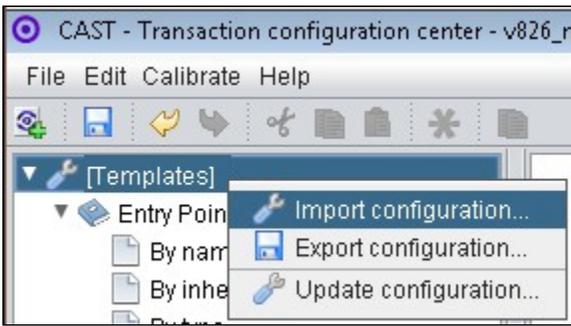
- Locate the .TCCSetup file in the extension folder: **Configuration\TCCDotNet_Silverlight.TCCSetup**
- In the CAST Transaction Configuration Center, ensure you have selected the **Templates** node:



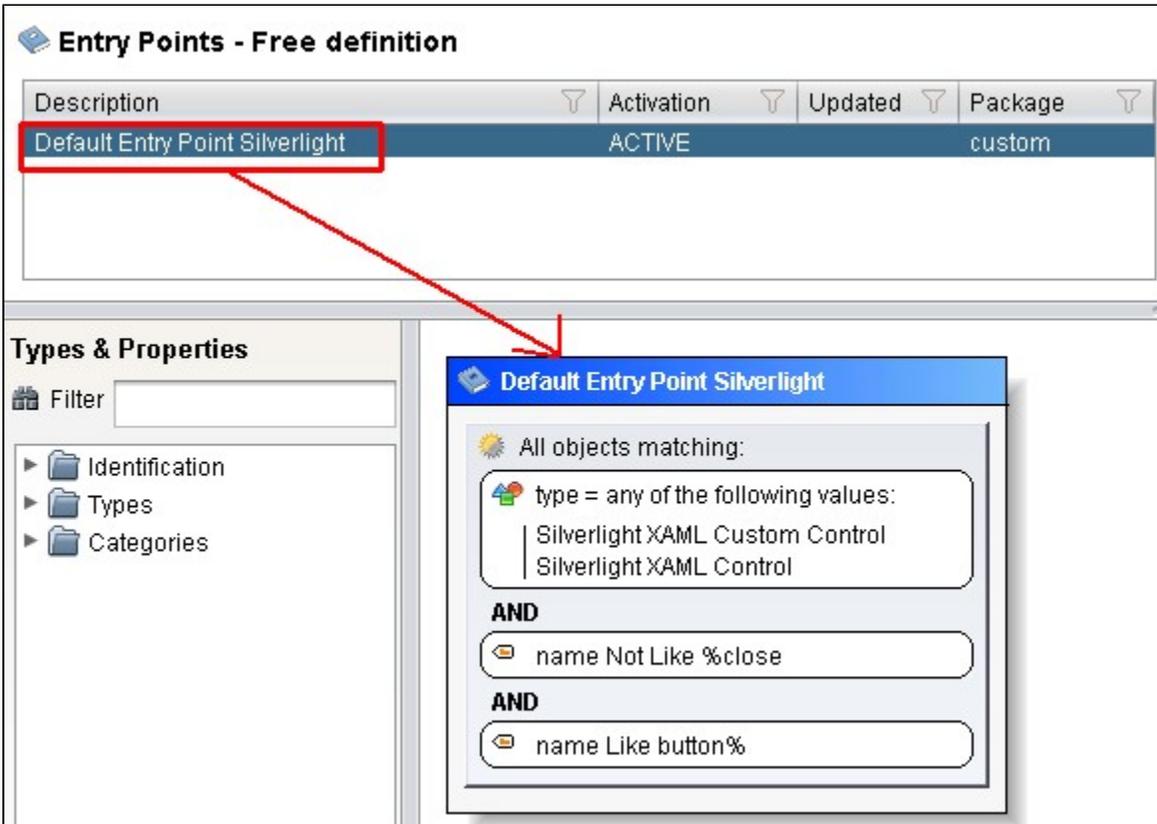
- This .TCCSetup file is to be imported into the CAST Transaction Calibration Center using either the:
 - **File > Import Configuration** menu option:



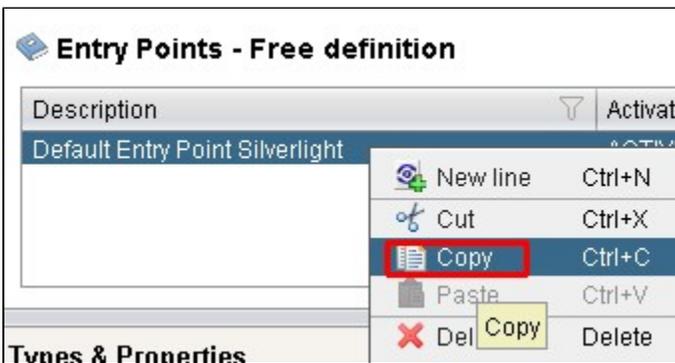
- Or right clicking on the **Template node** and selecting **Import Configuration**:



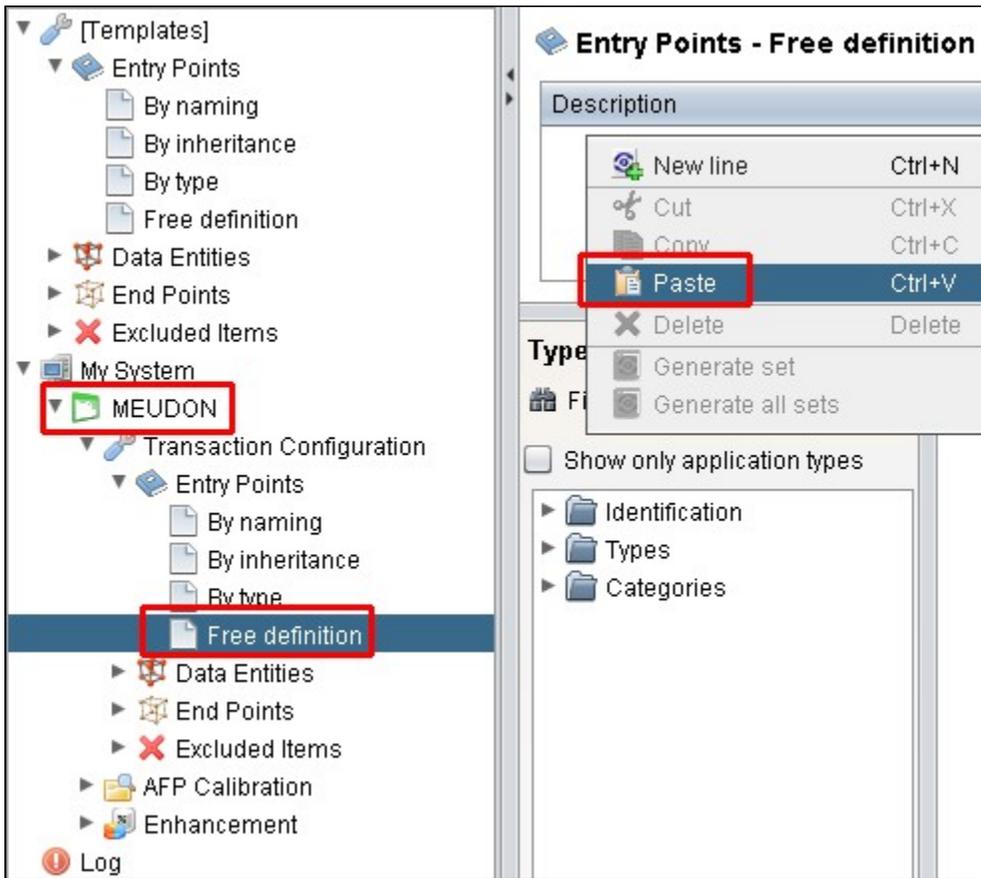
- The import of the "DotNet_Silverlight.TCCSetup" file will provide you with a sample Transaction Entry point in the **Free Definition** node under **Templates**:



- Now right click the "Default Entry Point Silverlight" item and select copy:



- Paste the item into the **equivalent node** under the **Application**, for example, below we have copied it into the **Application MEUDON**:



What results can you expect?

Once the analysis/snapshot generation has completed, you can view the results in the normal manner. The following **objects** and **links** will be displayed in CAST Enlighten:

Objects

All objects are represented under the **File browser** > **Xaml Source file** folders in CAST Enlighten:

Icon	Object type
	XAML Control

Note that if the **Name Attribute** of an object is present, then it will be used to display that object in CAST Enlighten. For example:

Name not present	Name present
 <p><u>Silverlight XAML Control</u> Button_1</p>	 <p><u>Silverlight XAML Control</u> Button_Nun</p>

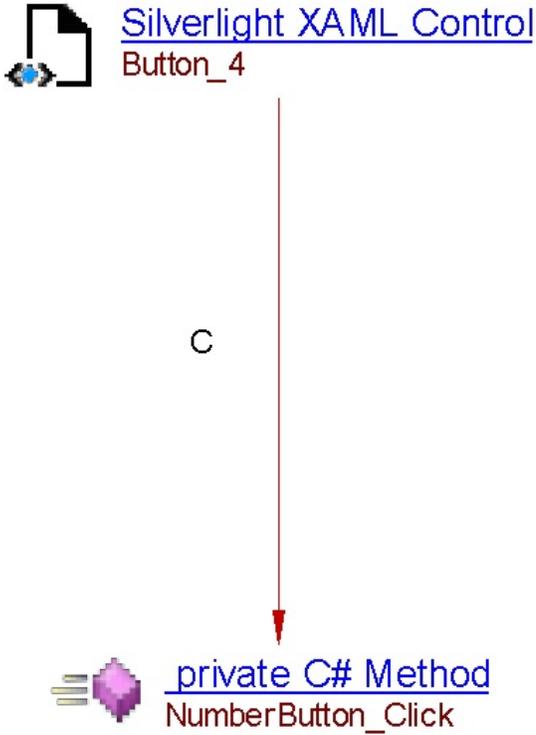
The following objects are detected:

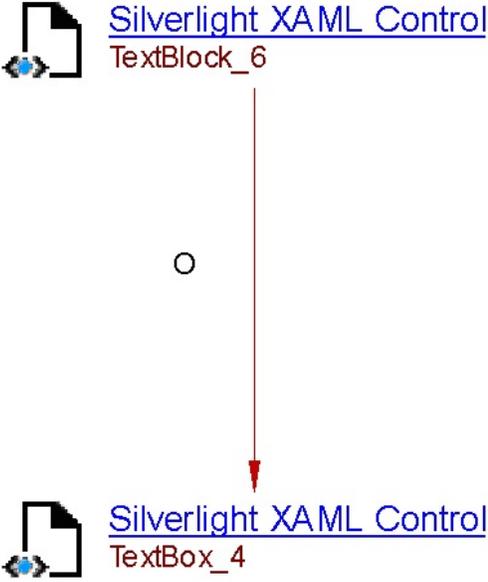
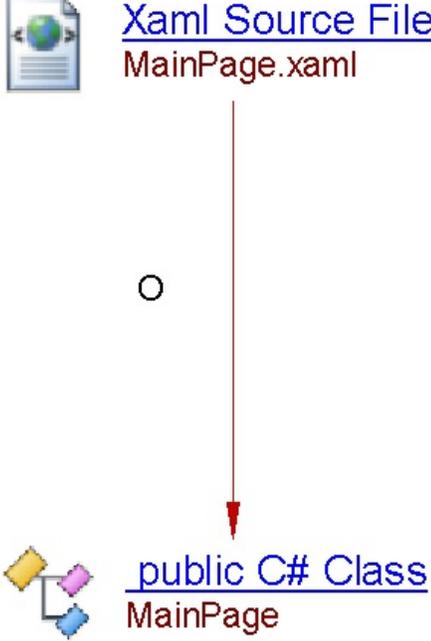
Border

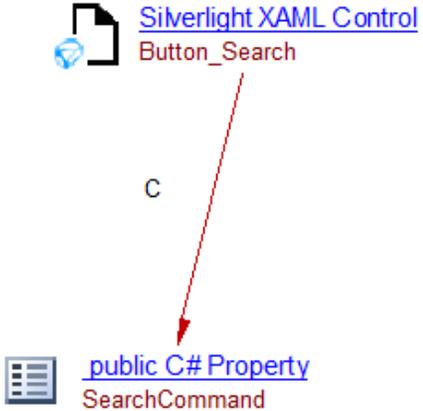
BulletDecorator
Button
Calendar
Canvas
CheckBox
ComboBox
ContextMenu
DataGrid
DatePicker
DockPanel
DocumentViewer
DropShadowBitmapEffect
Expander
FlowDocumentPageViewer
FlowDocumentReader
FlowDocumentScrollViewer
Frame
Grid
GridSplitter
GroupBox
HyperlinkButton
Image
Label
ListBox
ListView
Menu
Panel
PasswordBox
Popup
ProgressBar
PrintDialog
RadioButton
RepeatButton
Ribbon
RichTextBox
ScrollBar
ScrollViewer
Separator
Slider
StackPanel
StatusBar

TabControl
TextBlock
TextBox
ToolBar
ToolTip
TreeView
WrapPanel
Viewbox

Links

Source	Link type	Target	Example
XAML Control	call	Action Event method implemented in source file	 <p>The diagram illustrates a 'call' link. At the top, a document icon is next to the text 'Silverlight XAML Control' (blue) and 'Button_4' (red). A vertical red arrow points downwards from this text to a purple cube icon next to the text 'private C# Method' (blue) and 'NumberButton_Click' (red). The letter 'C' is centered between the two elements.</p>
XAML Control	relyOn	Set and get accessors of Property implemented in source file	 <p>The diagram illustrates a 'relyOn' link. At the top, a document icon is next to the text 'Silverlight XAML Control' (blue) and 'Menu_168' (red). Two red arrows originate from this text, each passing through a small circle. The left arrow points to a document icon next to the text 'public C# Property Get' (blue) and 'get' (red). The right arrow points to a document icon next to the text 'public C# Property Set' (blue) and 'set' (red).</p>

<p>XAML Control</p>	<p>relyOn</p>	<p>XAML Control which is used by another XAML Control</p>	 <p>The diagram shows a dependency between two Silverlight XAML controls. At the top, there is a document icon with a blue gear, followed by the text <u>Silverlight XAML Control</u> and <code>TextBlock_6</code>. A vertical red arrow points downwards from this control to another control below. In the middle of the arrow is a small black circle. The bottom control also has a document icon with a blue gear, followed by the text <u>Silverlight XAML Control</u> and <code>TextBox_4</code>.</p>
<p>XAML Source File</p>	<p>relyOn</p>	<p>Referred Class</p>	 <p>The diagram shows a dependency between an XAML source file and a C# class. At the top, there is a document icon with a globe, followed by the text <u>Xaml Source File</u> and <code>MainPage.xaml</code>. A vertical red arrow points downwards from this file to a class below. In the middle of the arrow is a small black circle. The bottom class has a class icon (three colored squares) followed by the text <u>public C# Class</u> and <code>MainPage</code>.</p>

C# property (relay command)	relyOn	C# method (delegate)	 <p>The diagram illustrates a dependency from a public C# Property named <code>SearchCommand</code> to a private C# Method named <code>OnSearch</code>. A red arrow points downwards from the property to the method, with a small circle containing the letter 'O' positioned in the middle of the arrow.</p>
XAML Control	call	C# property (relay command)	 <p>The diagram illustrates a dependency from a Silverlight XAML Control named <code>Button_Search</code> to a public C# Property named <code>SearchCommand</code>. A red arrow points downwards from the XAML control to the C# property, with a small circle containing the letter 'C' positioned in the middle of the arrow.</p>

Limitations

In this section we list the most significant functional limitations that may affect the analysis of applications using Silverlight:

- **Binding in code** is not supported - see [https://msdn.microsoft.com/en-us/library/ms742863\(v=vs.110\).aspx?cs-save-lang=1&cs-lang=csharp#code-snippet-1](https://msdn.microsoft.com/en-us/library/ms742863(v=vs.110).aspx?cs-save-lang=1&cs-lang=csharp#code-snippet-1) for an example
- **Binding links** are not handled correctly when the same **binding target** exists in different classes
- **The direction of the data flow** (Binding.Mode properties such as OneWay, TwoWay and OneWayToSource) is not supported - see [https://msdn.microsoft.com/en-us/library/ms752347\(v=vs.110\).aspx#direction_of_data_flow](https://msdn.microsoft.com/en-us/library/ms752347(v=vs.110).aspx#direction_of_data_flow) for an example
- **The RelativeSource property** (Gets or sets the binding source by specifying its location relative to the position of the binding target) is not supported - see [https://msdn.microsoft.com/en-us/library/system.windows.data.binding.relativesource\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.data.binding.relativesource(v=vs.110).aspx) for an example
- Some XAML control objects which are visible in CAST Enlighten may not be seen in the CAST Application Engineering Dashboard.
- Links between XAML Control objects and code embedded in XAML are not supported.