

Silverlight support for Csharp and VB.NET - 1.3



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.

- [What's new?](#)
 - [Funcrel](#)
 - [Beta 1](#)
 - [Alpha 2](#)
 - [Alpha 1](#)
- [Description](#)
 - [In what situation should you install this extension?](#)
- [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](#)
- [Packaging, delivering and analyzing your source code](#)
- [What results can you expect?](#)
 - [Objects](#)
 - [Links](#)
 - [Limitations](#)



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

What's new?

Funcrel

- APLSIL-72 - XAML source code files are now handled as transaction entry points, there after an upgrade to this release and the generation of a new snapshot on existing source code, you may find that your results have changed.

Beta 1

- Bug fixing, see: [Silverlight 1.3 - Bug Fix List](#)

Alpha 2

- Improvements have been made to the way exceptions are handled by the extension.

Alpha 1

- UserControls are now detected and saved as objects
- New BelongsTo link between UserControls and XAML Source Files
- Links will no longer be created from the XAML control to the Data Context.

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.

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:

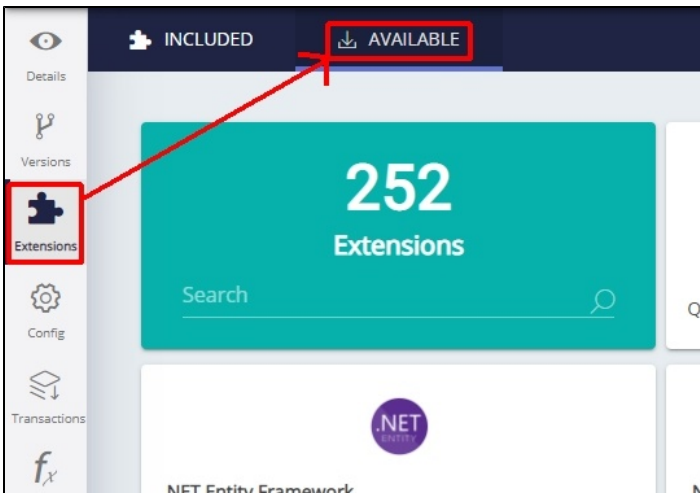
| DBMS | Supported |
|----------------------|-----------|
| CSS/PostgreSQL | ✓ |
| Oracle | ✓ |
| Microsoft SQL Server | ✗ |

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:

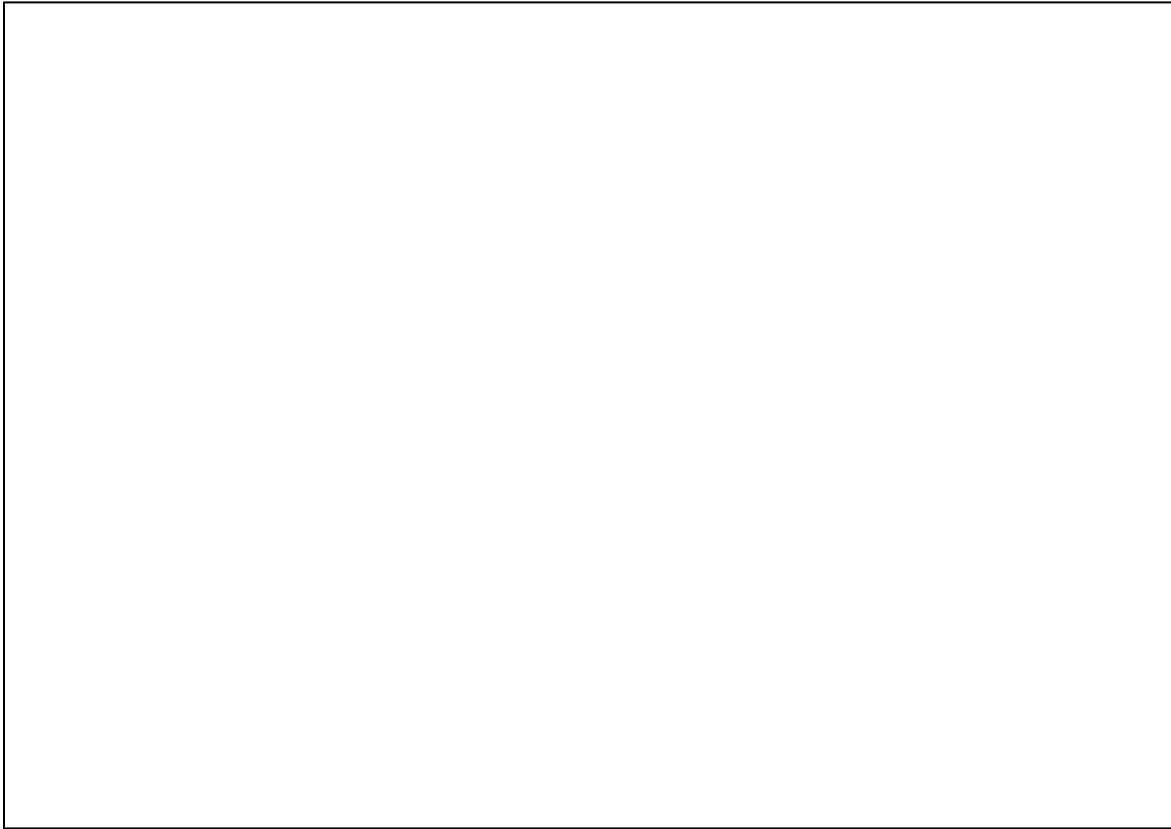


There is nothing further to do. Follow the instructions below to run a new analysis/snapshot to generate new results:

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

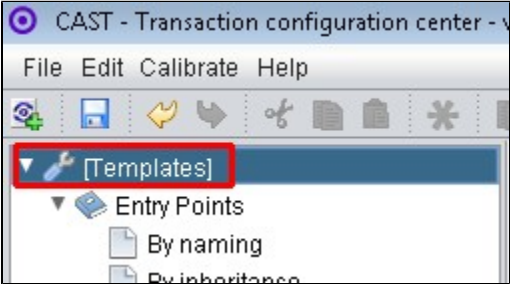
CAST Transaction Configuration Center (TCC) Entry Points

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:

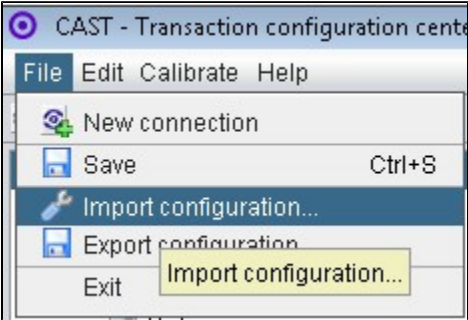


Manual import action for CAST AIP 8.2.x

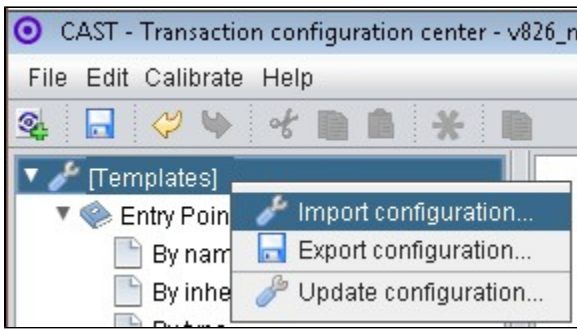
- Locate the .TCCSetup file in the extension folder: **Configuration\TCC\DotNet_Silverlight.TCCSetup**
- In the CAST Transaction Configuration Center, ensure you have selected the **Templates** node:



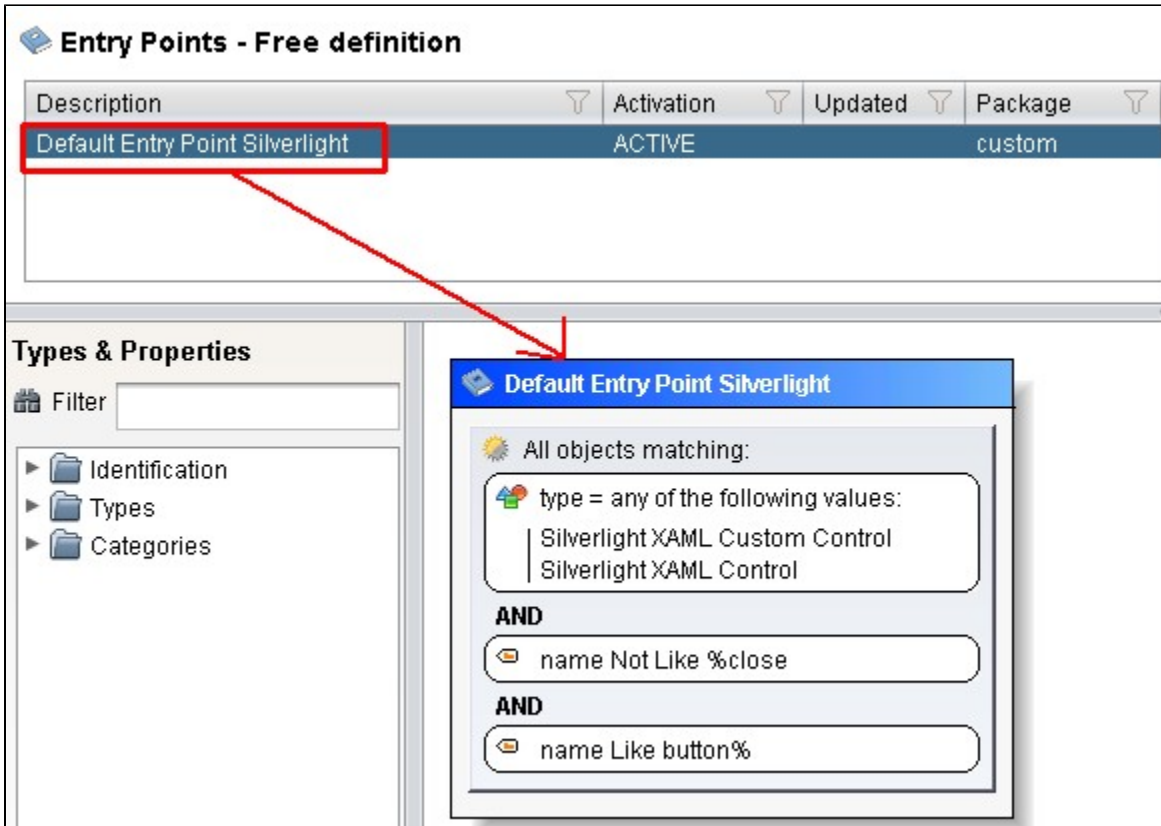
- This .TCCSetup file is to be imported into the CAST Transaction Calibration Center using either the:
 - **File > Import Configuration** menu option:



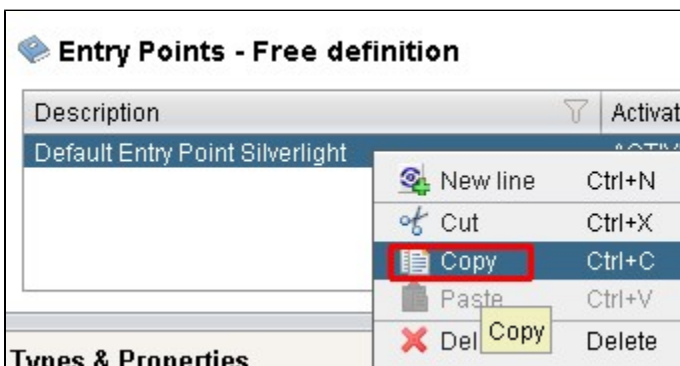
- Or right clicking on the **Template node** and selecting **Import Configuration**:



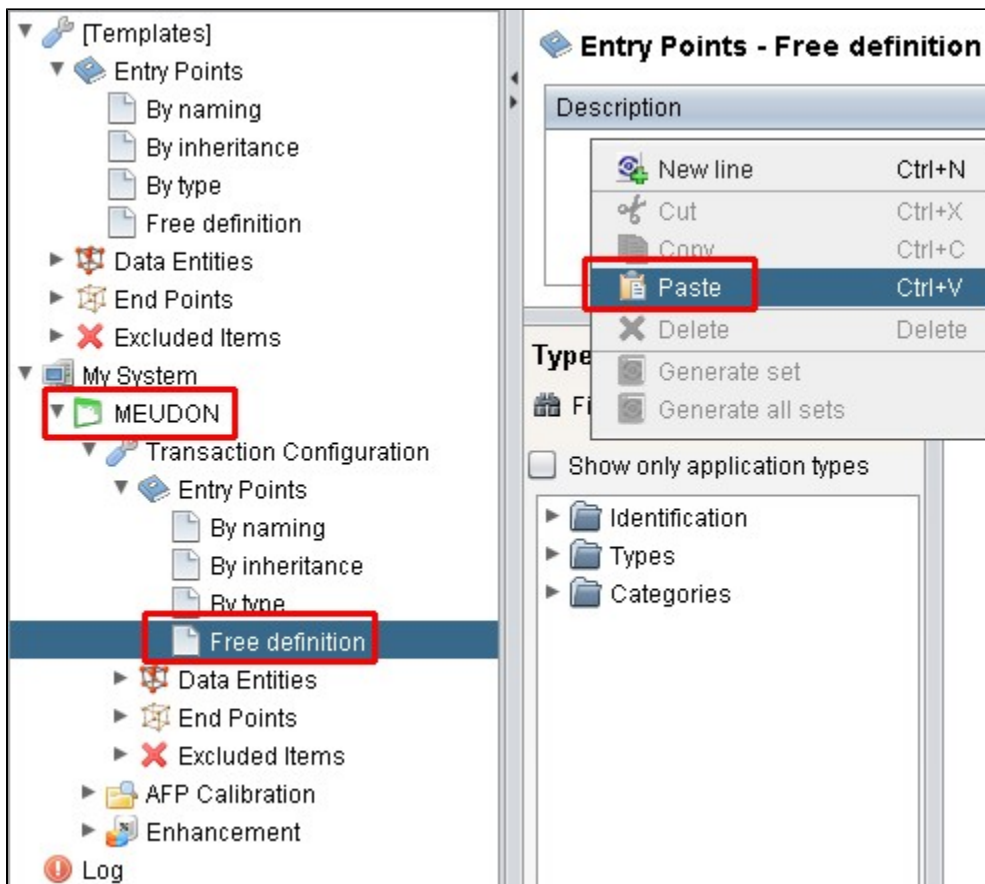
- The import of the "DotNet_Silverlight.TCCSetup" file will provide you with a sample Transaction Entry point in the **Free Definition** node under **Templates**:



- 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**:



Packaging, delivering and analyzing your source code

Once the extension is installed, no further configuration changes are required before you can package your source code and run an analysis. The process of packaging, delivering and analyzing your source code does not change in any way:


- **Package and deliver** your application (that includes source code which uses **Silverlight**) in the exact same way as you always have. You can refer to the existing official CAST documentation for more information about this - see: <http://doc.castsoftware.com/display/DOC83/Source+Code+Delivery+Guide+for+Application+Teams>.
- **Analyze** your delivered J2EE application source code in the CAST Management Studio in the exact same way as you always have - the source code which uses **Silverlight** will be detected and handled correctly. You can refer to the existing official CAST documentation for more information about configuring an analysis - see: <http://doc.castsoftware.com/display/DOC83/2.+Application+Analysis+Process+with+CAST+AIP>

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 |
|------------------|--------------|
| | |



Silverlight XAML Control Button_1



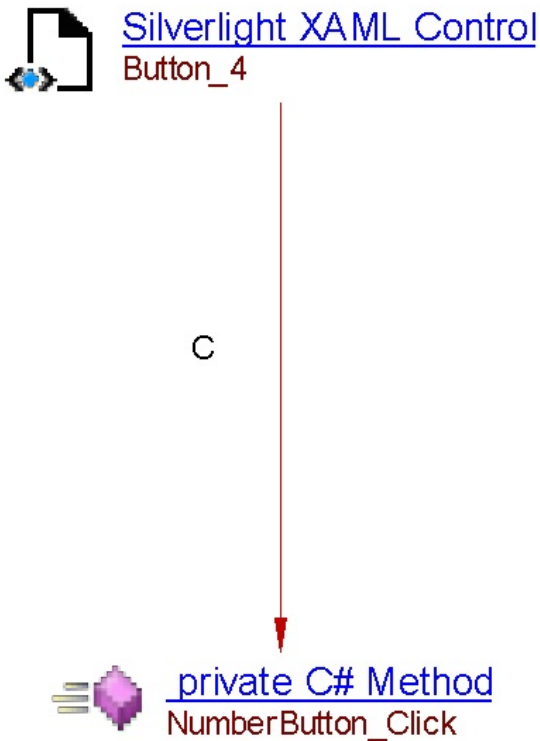
Silverlight Button_Nun

The following objects are detected:

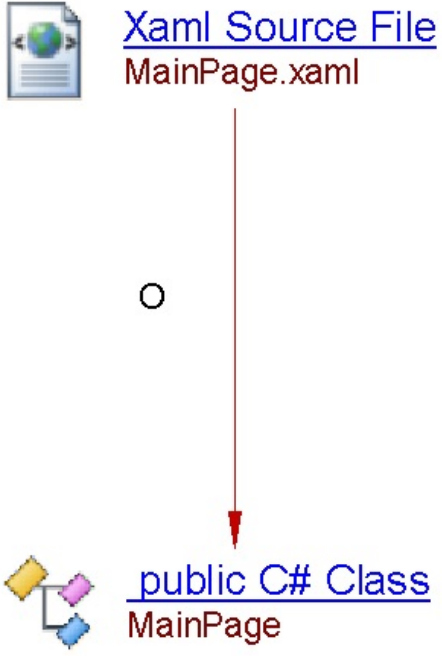
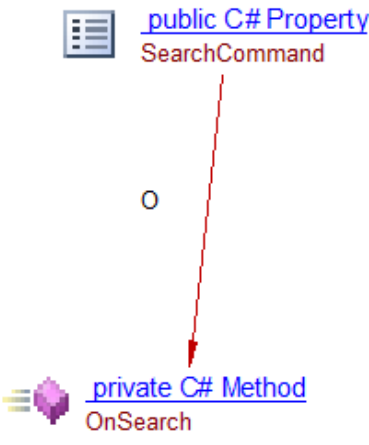
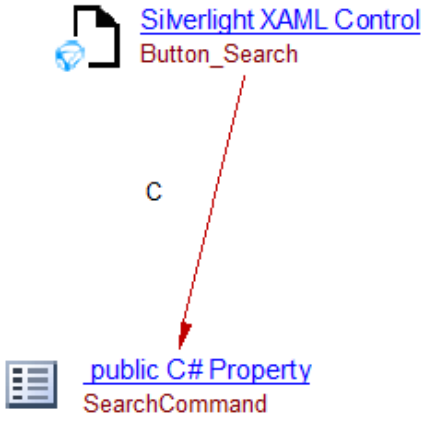
| |
|--------------------------|
| Border |
| BulletDecorator |
| Button |
| Calendar |
| Canvas |
| CheckBox |
| ComboBox |
| ContextMenu |
| DataGrid |
| DatePicker |
| DockPanel |
| DocumentViewer |
| DropShadowBitmapEffect |
| Expander |
| FlowDocumentPageViewer |
| FlowDocumentReader |
| FlowDocumentScrollViewer |
| 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 |
| UserControl |
| 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 between two code elements. At the top, there is a document icon with a gear, representing a Silverlight XAML Control, with the text "Silverlight XAML Control" and "Button_4" below it. A vertical red arrow points downwards from this element to a code icon (a purple diamond with horizontal lines), representing a private C# Method, with the text "private C# Method" and "NumberButton_Click" below it. The letter "C" is positioned in the middle of the arrow.</p> |

| | | | |
|---------------------|---------------|---|--|
| <p>XAML Control</p> | <p>relyOn</p> | <p>Set and get accessors of Property implemented in source file</p> | <pre> graph TD A["Silverlight XAML Control Menu_168"] --> B["_public C# Property Get get"] A --> C["_public C# Property Set set"] </pre> |
| <p>XAML Control</p> | <p>relyOn</p> | <p>XAML Control which is used by another XAML Control</p> | <pre> graph TD A["Silverlight XAML Control TextBox_6"] --> B["Silverlight XAML Control TextBox_4"] </pre> |
| <p>XAML Control</p> | <p>call</p> | <p>Binding property in DataContext</p> | <p>-</p> |

| | | | |
|------------------------------------|---------------|------------------------------------|--|
| <p>XAML Source File</p> | <p>relyOn</p> | <p>Referred Class</p> |  <p>The diagram shows a red arrow pointing from a document icon labeled "Xaml Source File MainPage.xaml" down to a class icon labeled "public C# Class MainPage". A small circle is positioned between the two elements.</p> |
| <p>C# property (relay command)</p> | <p>relyOn</p> | <p>C# method (delegate)</p> |  <p>The diagram shows a red arrow pointing from a list icon labeled "public C# Property SearchCommand" down to a method icon labeled "private C# Method OnSearch". A small circle is positioned between the two elements.</p> |
| <p>XAML Control</p> | <p>call</p> | <p>C# property (relay command)</p> |  <p>The diagram shows a red arrow pointing from a document icon labeled "Silverlight XAML Control Button_Search" down to a list icon labeled "public C# Property SearchCommand". A small circle is positioned between the two elements.</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.