

# [MS-SLXV]:

---

*Silverlight Xaml Vocabulary Specification 2008*

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications..
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). If you would prefer a written license, or if the language is not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Contents

[MS-SLXV]:.....	1
Silverlight Xaml Vocabulary Specification 2008.....	1
<b>Intellectual Property Rights Notice for Open Specifications Documentation.....</b>	<b>1</b>
Contents.....	2
Introduction .....	3
Specification Conventions.....	4
Xaml Types .....	4
Xaml Type Order .....	6
Xaml Members where [is attachable] is True .....	8
Xaml Members where [is event] is True .....	8
Xaml Members where [is static] is True.....	9
Constructors.....	9
The Silverlight Xaml Schema Information Set.....	10
<i>Document Constraints</i> .....	10
<i>Object Node Constraints</i> .....	11
<i>Member Constraints</i> .....	11
<i>Directives, XML Namespace</i> .....	11
<i>Intrinsic XamlType Information Items, XAML Namespace</i> .....	11
<i>Intrinsic XamlMember Information Items, XAML Namespace</i> .....	12
<i>Markup Compatibility</i> .....	12
<i>Dictionary Processing</i> .....	12
<i>Member Node Creation from Content</i> .....	12
Silverlight XamlType Information Items .....	14
Silverlight Xaml Text Syntax Information Sets .....	120
Appendix A:  References.....	133
Appendix B:  Microsoft Silverlight Behavior.....	134
Index.....	135

## Introduction

Xaml is defined in the Xaml Object Mapping Specification (referred to as '[MS-XAML](#)' for short). The MS-XAML specification enables each application that uses Xaml to define its own vocabulary. Vocabularies are formally specified using the Xaml Schema Information Set, a data model defined by MS-XAML.

This specification defines the Xaml Schema Information Set information items for the Silverlight Xaml Vocabulary. The information items in the Xaml Schema Information Set presented in this specification can be used in conjunction with MS-XAML to determine whether any particular Xaml instance or XML document is valid Silverlight Xaml.

## Specification Conventions

The MS-XAML specification only defines a data model for the Xaml Schema Information Set. It does not prescribe the representation of the information items that constitute a schema. While MS-XAML does introduce a notation with which it defines its intrinsic information items, this specification does not use that notation. Instead, a more compact representation is used to minimize redundancy.

The Silverlight Xaml Vocabulary contains several hundred information items, many of which share much in common. The following sections describe the conventions used in this specification, which exploit this commonality to reduce the volume of text required to describe each information item, and to make it easier to see each item's distinguishing features.

## Xaml Types

The MS-XAML specification defines a XamlType Information Item. Throughout this specification, when a XamlType Information Item is presented, only those values that do not match the default value are listed. The following table defines the default values for a XamlType Information Item.

Property	Default Value
[is default constructible]	True
[is nullable]	True
[text syntax]	Null
[dictionary key property]	Null
[name property]	Null
[xml lang property]	Null
[trim surrounding whitespace]	False
[whitespace significant collection]	False
[is list]	False
[is dictionary]	False
[members]	Empty set
[content property]	Null
[allowed types]	Empty set
[allowed key types]	Empty set
[is xdata]	False
[is name scope]	False
[constructors]	Empty set
[return value type]	Null

This specification denotes non-default values for the information set items with rows that begin with the relevant property name in square brackets. Each Xaml Type definition in this specification includes some non-normative information, in order to aid understanding of the type. In order to distinguish them, these rows begin with a name in parentheses.

There is a (usage) row that illustrates the way in which the type is used in XML. If the type cannot contain content, a self-closing tag will be displayed, as the following example shows:

---

**(usage)** `<DependencyObject />`

---

Some types offer a choice of content. The following example indicates that the three fixed string values shown are acceptable as content:

---

**(usage)** `OnLastWindowClose | OnMainWindowClose | OnExplicitShutdown`

---

Elements that may contain string content look like the following example. Note that there are typically constraints on which strings are valid. The purpose of the (usage) row is only to provide an approximate indication of content, rather than a formal description.

---

**(usage)** `<PropertyPath> string </PropertyPath>`

---

Types that can contain an instance of some other type show the name of the type in the usage, as shown in the following example:

---

**(usage)** `<BeginStoryboard> Storyboard </BeginStoryboard>`

---

Types that can contain multiple instances of some other type indicate this with an asterisk, as shown in the following example:

---

**(usage)** `<XmlNamespaceMappingCollection> XmlNamespaceMapping* </XmlNamespaceMappingCollection>`

---

Some types cannot be used directly in a Xaml document. For example, a type might be used as the [value type] of a member, but may be marked as [is default constructible] False, and have no [text syntax]. A Xaml document might contain types that list such a type in their [types assignable to] property, but the type itself cannot be used directly. (This corresponds to the idea of an abstract type in some object-oriented programming systems.) Such types have “None” in their Usage row.

Type definitions also include (description) rows, both for the type and also for the members of that type. This is also non-normative.

The (used by) row is also non-normative. Each (used by) row provides a list of types that use this type in some way. For example, it lists types that use this type as the [value type] of a member. The (used by) row is provided to make it easier to see how a type is used, and it does not translate into a property in the Xaml Schema Information Set.

Some types that are used as the [value type] of a member, but which are not typically used directly as objects have a (types assignable from) row. This is non-normative, and is provided as a guide to the role of the type. Note that this list is necessarily not complete, because the set of types from which a type is assignable is not closed: anyone is free to define a new Xaml vocabulary which defines types that are assignable to types in this specification.

## Xaml Type Order

Xaml types in this specification are ordered in a hierarchical way. The Silverlight Xaml Vocabulary uses the [types assignable to] property in a way that corresponds to inheritance in object-oriented programming. ‘Base’ types are presented before ‘derived’ types. (That is, most types are defined before appearing in the [types assignable to] lists of other types.)

The number that appears to the left of the first row of each XamlType Information Item indicates its position in the inheritance hierarchy of the types. The following example shows the XamlType Information Items for the Fruit, Apple, and Banana types. The numbers indicate that the Apple and Banana types derive from Fruit.

---

### 4.5 Fruit

(usage)	Value
...	...
property <i>N</i>	Value

---

#### 4.5.1 Apple

(usage)	Value
...	...
property <i>N</i>	Value

---

#### 4.5.2 Banana

(usage)	Value
...	...
property <i>N</i>	Value

Since this specification models typical object-oriented inheritance, a ‘derived’ type inherits all members from a ‘base’ type. This is not made explicit. For each type, only additional members are listed. The MS-XAML specification does not require this inheritance-like style. In the Xaml Schema Information Set data model, each type lists its members exhaustively. Therefore, the correct interpretation of a type definition in this specification is that the corresponding XamlType Information Item’s [members] property should include not just the listed members, but also all of the [members] of each type listed in its [types assignable to] property.

XamlMember Information Items have numerous properties, and in this specification, members are more similar than they are different. So a notation is used to minimize redundancy. Some XamlMember Information Item properties may be omitted. Unless specified otherwise, the default values described in the following table apply.

Property	Default Value
[text syntax]	Null
[is read only]	False
[is static]	False
[is attachable]	False
[target type]	Null
[allowed location]	Any
[is event]	False
[is directive]	False

Members are not defined in distinct sections of this specification – they are listed inside their defining type following a row named (properties). This means that the [owner type] member defined by MS-XAML is never specified explicitly in this specification. The [owner type] is always the type in which the member definition appears. Likewise, the [members] property of the defining type is never explicitly defined – it always contains all of the members listed for that type. The [name] and [value type] are specified on the first line of the property description. This line may be followed by non-default values for other properties. The following example shows the XamlType Information Item for the Satsuma type, which defines a member named SegmentCount of type Int32.

## 4.5.9 Satsuma

(usage)	<Satsuma />
(description)	Specifies a small, orange citrus fruit.
(types assignable to)	Satsuma Fruit Object
(properties)	
SegmentCount	Int32
(description)	The number of segments in this satsuma.

If all of the XamlMember Information Item properties had been listed in full for this property, it would look like the following table.

Property	Value
[name]	SegmentCount
[owner type]	Satsuma
[value type]	Int32
[text syntax]	Null
[is read only]	False
[is static]	False

[is attachable]	False
[target type]	Null
[allowed location]	Any
[is event]	False

As with the type-level (description), the per-member (description) entries in this specification are non-normative.

XamlType Information Item descriptions in this document may contain up to three additional member categories: attachable members, event members, and static members. These three member categories have slightly different defaults, and are grouped separately in the type definitions for clarity. The conventions for these member categories are defined in the following sections.

### Xaml Members where [is attachable] is True

A type that defines members whose [is attachable] property is True will list them in a section that begins with “(attachable properties)”. The following FruitBowl type example defines a Children member for which the normal defaults apply. This FruitBowl type also defines an attachable member named ZIndex for which [is attachable] is True. (The other member defaults still apply for ZIndex.)

## 4.12 FruitBowl

(usage)	<FruitBowl>Fruit*</FruitBowl>
(description)	A container of fruit.
[types assignable to]	FruitBowl Object
[content property]	Children
(properties)	
Children	FruitCollection
(description)	The items of fruit
(attachable properties)	
FruitBowl.ZIndex	Int32
(description)	Indicates how deeply buried within the fruit bowl a piece of fruit is.
[target type]	Fruit

The name for an attachable member is specified as *TypeName.MemberName*. This is a syntactical convention to make it clear that this is an attachable property, and to illustrate how the property will look in a Xaml document. The [name] property of the XamlMember Information Item will only contain the *MemberName* part (the part after the period).

### Xaml Members where [is event] is True

A type that defines members whose [is event] property is True will list them in a section that begins with “(events)”. For members in this section, the default value for [value type] is the x:XamlEvent type defined in Section 5 “Intrinsic Schema Information Items” of MS-XAML. (The other defaults still apply.)



## Xaml Members where [is static] is True

A type that defines members whose [is static] property is True will list them in a section that begins with “(static properties)”.

### Constructors

Types with a non-empty [constructors] property use a convention similar to that for members. This example includes a constructor:

## 4.97.4 BruleeToppingExtension

<b>(usage)</b>	{BruleeToppingExtension}
<b>(description)</b>	Defines the way in which the topping on a brulée-style dessert is prepared.
<b>[types assignable to]</b>	Brulee MarkupExtension Object
<b>(used by)</b>	FruitBrulee CremeBrulee
<b>[return value type]</b>	BruleeTopping
<b>[constructors]</b>	
<b>(2 parameters)</b>	
<b>thickness</b>	Double
<b>(description)</b>	The thickness of the topping in 1/96 <sup>th</sup> of an inch.
<b>burnFactor</b>	Double
<b>(description)</b>	The extent to which the sugar is burnt: 0 for raw sugar, 1 for carbon.

Unlike members, which are identified by names, a constructor for a type is distinguished only by the number of parameters it has. This type has a single constructor that takes two parameters. This is interpreted as a Constructor Information Item, whose [arguments] contains one entry per parameter. The [arguments] list contains just XamlType Information Items – both the Double type in this case. Note that the type is the only formal part of the constructor argument – the name and description are only provided for informational purposes.

## The Silverlight Xaml Schema Information Set

The Silverlight Xaml Schema Information Item is a Xaml Schema Information Item (as defined in Section 3 “Xaml Schema Information Set” of MS-XAML). Its properties are defined in the following table.

Property	Value
[target namespace]	“http://schemas.microsoft.com/winfx/2006/xaml/presentation”
[types]	All of the XamlType Information Items defined in the “Silverlight XamlType Information Items” section of this specification.
[assignable types]	All of the XamlType Information Items defined in the “ <b>Error! Reference source not found.</b> ” section of this specification.
[directives]	The XamlMember Information Items defined in this specification for which the [is directive] property is True.
[compatible with schemas]	“http://schemas.microsoft.com/client/2007”

A Silverlight Xaml instance **MUST** be well-formed and valid with respect to this schema, using the rules for ‘well-formed’ and ‘valid’ defined in MS-XAML, except as noted in [Silverlight Exceptions to \[MS-XAML\] Specification](#).

An XML document that is a Silverlight Xaml document **MUST** yield a Silverlight Xaml instance when the processing rules in Section 6 “Creating a Xaml Information Set from XML” are applied using this schema, except as noted in [Silverlight Exceptions to \[MS-XAML\] Specification](#).

## Silverlight Exceptions to [MS-XAML] Specification

Valid Silverlight Xaml Documents include the set of XamlTypes defined in this specification. The rules for validity of a Xaml document for Silverlight that will create a Xaml Infoset are more constrained than the combination of the XamlTypes plus the specification material in [MS-XAML]. Also, Silverlight’s technique for mapping XML into a Xaml Infoset does not completely conform to Section 6 of [MS-XAML]. The following section documents the variances in rules for validation and Infoset processing of a Silverlight Xaml document.

Important: All material in this section applies specifically to Silverlight 2008 Xaml documents.

### *Document Constraints*

To be considered a valid Silverlight Xaml document, and to create a valid Silverlight Infoset from that document, Silverlight imposes the following additional constraints on setting xmlns values on the Document Information Item. The following are constraints to add to section 4.1.1 of [MS-XAML]:

- 4.1.1.S1. The root element **MUST** contain a default xmlns declaration; no implicit value is assumed.
- 4.1.1.S2. The default xmlns declaration on the root element **MUST** be one of the following:  
http://schemas.microsoft.com/winfx/2006/xaml/presentation, http://schemas.microsoft.com/client/2007.

## Object Node Constraints

The following is a constraint to add to section 4.2.1 of [MS-XAML]. Silverlight imposes the following additional constraint on setting xmlns values on an Object Node Information Item.

4.2.1.S1. If a default xmlns declaration is made on an Object Node Information Item, that default xmlns declaration MUST be one of the following: <http://schemas.microsoft.com/winfx/2006/xaml/presentation>, <http://schemas.microsoft.com/client/2007>.

## Member Constraints

Silverlight imposes the following special constraint on **UserControl**. The following should be considered an additional constraint for section 4.3.2 of [MS-XAML].

4.3.2.S1. A Silverlight **UserControl** MUST specify the x:Class directive if it contains content, and/or sets its **Content** member. (**Note:** The x:Class directive requirement effectively dictates that the **UserControl** be at the root of the document.)

## Directives, XML Namespace

6.6.5.S1. xml:space is valid in a Silverlight Xaml document, but information conveyed by xml:space is not applied to member nodes during creation from an XML element.

5.3.15.S1. xml:base MUST NOT exist as a member node in a Silverlight Xaml Infoset. A document that contains xml:base as a member node is invalid.

## Intrinsic XamlType Information Items, XAML Namespace

6.6.2.S1. Valid Silverlight Xaml Documents may not contain x:NullExtension usages of the form: attribute usage of {x:NullExtension}, or element usage of either <x:Null /> or <x:NullExtension />.

Specifically, for x:NullExtension, the following two bullet points do not apply:

- If `schema(xmlObjectElement)[types]` contains a type 't' where `t[name]` matches `xmlObjectElement[local name]`, let `objectType` be that t.
- Otherwise, if `schema(xmlObjectElement)[types]` contains a type 't' where `t[name]` matches the concatenation of `xmlObjectElement[local name]` and 'Extension' and `t[types assignable to]` contains the x:MarkupExtension type (5.2.21), let `objectType` be that t.

5.2.1.S1. Valid Silverlight Xaml Documents may not contain x:ArrayExtension.

5.2.2.S1. Valid Silverlight Xaml Documents may not contain x:StaticExtension.

5.2.3.S1. Valid Silverlight Xaml Documents may not contain x:TypeExtension.

5.2.22.S1. Valid Silverlight Xaml Documents may not contain x:Code.

5.2.23.S1. Valid Silverlight Xaml Documents may not contain x:XData.

### *Intrinsic XamlMember Information Items, XAML Namespace*

- 5.3.5.S1. Valid Silverlight Xaml Documents may not contain x:Uid.
- 5.3.8.S1. Valid Silverlight Xaml Documents may not contain x:Subclass.
- 5.3.9.S1. Valid Silverlight Xaml Documents may not contain x:ClassModifier.
- 5.3.10.S1. Valid Silverlight Xaml Documents may not contain x:FieldModifier.
- 5.3.11.S1. Valid Silverlight Xaml Documents may not contain x:TypeArguments.

### *Markup Compatibility*

6.3.S1. Markup compatibility refers to the namespace <http://schemas.openxmlformats.org/markup-compatibility/2006>. The only valid markup compatibility construct in a Silverlight Xaml Infoset is Ignorable. If used, Ignorable MUST be an attribute usage, and MUST be applied on the root element. Element usage or attribute usage on non-root elements are each invalid.

### *Dictionary Processing*

4.3.1.4.S1. The addition of the bold bullet item below into [MS-XAML] section 4.3.1.4 correctly represents the behavior of Silverlight Xaml Document processing.

- Each Object Node Information Item ‘dictItem’ in [values] MUST match at least one of the following (and let the first of these rules that matches define keyMemberNode for that dictItem):
  - dictItem[member nodes] contains a Member Node Information Item that is the x:Key directive (defined in 5.3.5).
  - **dictItem[member nodes] contains a Member Node Information Item that is the x:Name directive or the [name property].**
  - dictItem[member nodes] contains a Member Node Information Item whose [member] is dictItem[type][dictionary key property].

4.3.1.4.S2. The string used as the key MUST conform to the XamlName Grammar as defined by [MS-XAML].

### *Member Node Creation from Content*

Silverlight has variances in member node creation from content, in cases where the content is provided as x:String.

6.6.6.S1. Per [MS-XAML], XamlTypes that are a singular type where x:String exists in [assignable to] should be processed as string content for a [content property]. Silverlight Xaml Documents are invalid if an x:String is found in this location. For example, <Button>hello world</Button> is not allowed in a valid Silverlight Xaml Document to set Button’s property named “Content”, even though the XamlType of the [content property] (“Content”) for **Button** is x:Object and is thus assignable to x:String.

6.6.6.S2. Per [MS-XAML], XamlTypes that are a collection type, where a type that is x:Object exists in [allowed types], should be processed as string content for a [content property]. Silverlight Xaml Documents are invalid if an x:String found in this location. For example, <ItemsControl>Item1</ItemsControl> is not allowed in a valid Silverlight Xaml Document to set the ItemsControl's property named "Items", even though the XamlType of the [content property] for **ItemsControl** ("Items") is x:Object and is thus assignable from x:String.

## Silverlight XamlType Information Items

---

### 1 x:Object

link to externally defined type [x:Object, from \[MS-XAML\]](#)

#### 1.1 AlignmentX

(usage)	<a href="#">Left</a>   <a href="#">Center</a>   <a href="#">Right</a>
(description)	Describes how content is positioned horizontally in a container.
[types assignable to]	<a href="#">AlignmentX x:Object</a>
(used by)	<a href="#">TileBrush</a>
[is nullable]	False
[text syntax]	<a href="#">AlignmentXSyntax</a>

#### 1.2 AlignmentY

(usage)	<a href="#">Top</a>   <a href="#">Center</a>   <a href="#">Bottom</a>
(description)	Describes how content is positioned vertically in a container.
[types assignable to]	<a href="#">AlignmentY x:Object</a>
(used by)	<a href="#">TileBrush</a>
[is nullable]	False
[text syntax]	<a href="#">AlignmentYSyntax</a>

#### 1.3 Application

(usage)	<Application />
(description)	Encapsulates a Silverlight application.
[types assignable to]	<a href="#">Application x:Object</a>
(used by)	<a href="#">Application</a>

<b>(properties)</b>	
<b>RootVisual</b>	<a href="#">UIElement</a>
<b>(description)</b>	
The main application UI.	
<b>(events)</b>	
<b>Exit</b>	Occurs just before an application shuts down and cannot be canceled.
<b>Startup</b>	Occurs when an application is started.
<b>UnhandledException</b>	Occurs when an exception that is raised by Silverlight is not handled.

## 1.4 x:Array

**link to externally defined type** [x:Array, from \[MS-XAML\]](#)

## 1.5 AssemblyPartCollection

<b>(usage)</b>	<AssemblyPartCollection> <a href="#">AssemblyPart</a> * </AssemblyPartCollection>
<b>(description)</b>	Stores a collection of AssemblyPart objects. Provides collection support for the Parts property.
<b>[types assignable to]</b>	<a href="#">AssemblyPartCollection</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Deployment</a>
<b>[is list]</b>	True
<b>[allowed types]</b>	<a href="#">AssemblyPart</a>

## 1.6 AutomationProperties

<b>(usage)</b>	None.
<b>(description)</b>	Provides a means of getting or setting the value of the associated properties of the instance of the AutomationPeer element.
<b>[types assignable to]</b>	<a href="#">AutomationProperties</a> <a href="#">x:Object</a>
<b>(attachable properties)</b>	
<b>AutomationProperties. AcceleratorKey</b>	<a href="#">x:String</a>

(description)	The accelerator key for the element.
[target type]	<a href="#">DependencyObject</a>
<b>AutomationProperties.AccessKey</b>	<a href="#">x:String</a>
(description)	The access key for the element.
[target type]	<a href="#">DependencyObject</a>
<b>AutomationProperties.AutomationId</b>	<a href="#">x:String</a>
(description)	The string that uniquely identifies the element to UI Automation.
[target type]	<a href="#">DependencyObject</a>
<b>AutomationProperties.HelpText</b>	<a href="#">x:String</a>
(description)	The help text for the element.
[target type]	<a href="#">DependencyObject</a>
<b>AutomationProperties.IsRequiredForForm</b>	<a href="#">x:Boolean</a>
(description)	A value that indicates whether the element is required to be filled out on a form.
[target type]	<a href="#">DependencyObject</a>
<b>AutomationProperties.ItemStatus</b>	<a href="#">x:String</a>
(description)	A description of the status of an item in an element.
[target type]	<a href="#">DependencyObject</a>
<b>AutomationProperties.ItemType</b>	<a href="#">x:String</a>
(description)	A description of the type of the specified element.
[target type]	<a href="#">DependencyObject</a>
<b>AutomationProperties.LabeledBy</b>	<a href="#">UIElement</a>
(description)	The element that contains the text label for the element.
[target type]	<a href="#">DependencyObject</a>
<b>AutomationProperties.Name</b>	<a href="#">x:String</a>
(description)	The automation name of the element.
[target type]	<a href="#">DependencyObject</a>



## 1.7 Binding

<b>(usage)</b>	<Binding />
<b>(description)</b>	Defines a binding that connects the properties of binding targets and data sources.
<b>[types assignable to]</b>	<a href="#">Binding</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>Converter</b>	<a href="#">IValueConverter</a>
<b>(description)</b>	The converter object that is called by the binding engine to modify the data as it is passed between the source and target, or vice versa.
<b>ConverterCulture</b>	<a href="#">CultureInfo</a>
<b>(description)</b>	The culture to be used by the Converter.
<b>ConverterParameter</b>	<a href="#">x:Object</a>
<b>(description)</b>	A parameter that can be used in the Converter logic.
<b>Mode</b>	<a href="#">BindingMode</a>
<b>(description)</b>	A value that indicates the direction of the data flow in the binding.
<b>NotifyOnValidationError</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether or not the <a href="#">BindingValidationError</a> event is raised on validation errors.
<b>Path</b>	<a href="#">PropertyPath</a>
<b>(description)</b>	The path to target property of the binding.
<b>Source</b>	<a href="#">x:Object</a>
<b>(description)</b>	The data source for the binding.
<b>ValidatesOnExceptions</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether the binding engine will report exception validation errors.

## 1.8 BindingMode

<b>(usage)</b>	<a href="#">OneWay</a>   <a href="#">OneTime</a>   <a href="#">TwoWay</a>
<b>(description)</b>	Describes how the data propagates in a binding.
<b>[types assignable to]</b>	<a href="#">BindingMode</a> <a href="#">x:Object</a>

(used by)	<a href="#">Binding</a>
[is nullable]	False
[text syntax]	<a href="#">BindingModeSyntax</a>

## 1.9 x:Boolean

link to externally defined type	<a href="#">x:Boolean, from [MS-XAML]</a>
---------------------------------	---

## 1.10 BrushMappingMode

(usage)	<a href="#">Absolute</a>   <a href="#">RelativeToBoundingBox</a>
(description)	Specifies the coordinate system used by a Brush.
[types assignable to]	<a href="#">BrushMappingMode</a> <a href="#">x:Object</a>
(used by)	<a href="#">GradientBrush</a>
[is nullable]	False
[text syntax]	<a href="#">BrushMappingModeSyntax</a>

## 1.11 x:Byte

link to externally defined type	<a href="#">x:Byte, from [MS-XAML]</a>
---------------------------------	--

## 1.12 x:Char

link to externally defined type	<a href="#">x:Char, from [MS-XAML]</a>
---------------------------------	--

## 1.13 ClickMode

<b>(usage)</b>	<a href="#">Release</a>   <a href="#">Press</a>   <a href="#">Hover</a>
<b>(description)</b>	Specifies when the Click event should be raised for a control.
<b>[types assignable to]</b>	<a href="#">ClickMode x:Object</a>
<b>(used by)</b>	<a href="#">ButtonBase</a>
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">ClickModeSyntax</a>

## 1.14 Color

<b>(usage)</b>	<a href="#">Black</a>   <a href="#">Blue</a>   <a href="#">Brown</a>   <a href="#">Cyan</a>   <a href="#">DarkGray</a>   <a href="#">Gray</a>   <a href="#">Green</a>   <a href="#">LightGray</a>   <a href="#">Magenta</a>   ...
<b>(description)</b>	Describes a color in terms of alpha, red, green, and blue channels.
<b>[types assignable to]</b>	<a href="#">Color x:Object</a>
<b>(used by)</b>	<a href="#">ColorKeyFrame</a> <a href="#">Colors</a> <a href="#">DrawingAttributes</a> <a href="#">GradientStop</a> <a href="#">SolidColorBrush</a>
<b>[is default constructible]</b>	False
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">ColorSyntax</a>
<b>(properties)</b>	
<b>A</b>	<a href="#">x:Byte</a>
<b>(description)</b>	The sRGB alpha channel value of the color.
<b>B</b>	<a href="#">x:Byte</a>
<b>(description)</b>	The sRGB blue channel value of the color.
<b>G</b>	<a href="#">x:Byte</a>
<b>(description)</b>	The sRGB green channel value of the color.
<b>R</b>	<a href="#">x:Byte</a>
<b>(description)</b>	The sRGB red channel value of the color.

## 1.15 ColorInterpolationMode

<b>(usage)</b>	<a href="#">ScRgbLinearInterpolation</a>   <a href="#">SRgbLinearInterpolation</a>
<b>(description)</b>	Determines how the colors in a gradient are interpolated.

[types assignable to]	<a href="#">ColorInterpolationMode</a> <a href="#">x:Object</a>
(used by)	<a href="#">GradientBrush</a>
[is nullable]	False
[text syntax]	<a href="#">ColorInterpolationModeSyntax</a>

## 1.16 ColorKeyFrameCollection

(usage)	<ColorKeyFrameCollection> <a href="#">ColorKeyFrame</a> * </ColorKeyFrameCollection>
(description)	Represents a collection of ColorKeyFrame objects that can be individually accessed by index.
[types assignable to]	<a href="#">ColorKeyFrameCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">ColorAnimationUsingKeyFrames</a>
[is list]	True
[allowed types]	<a href="#">ColorKeyFrame</a>

## 1.17 Colors

(usage)	{x:Static <a href="#">Colors.StaticPropertyName</a> }
(description)	Implements a set of predefined colors.
[types assignable to]	<a href="#">Colors</a> <a href="#">x:Object</a>

## 1.18 ColumnDefinitionCollection

(usage)	None.
(description)	Provides access to an ordered, strongly typed collection of ColumnDefinition objects.
[types assignable to]	<a href="#">ColumnDefinitionCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">Grid</a>
[is list]	True

[allowed types]	<a href="#">ColumnDefinition</a>
-----------------	----------------------------------

## 1.19 CornerRadius

(usage)	<nerRadius> string </nerRadius>
---------	---------------------------------

(description)	Describes the characteristics of a rounded corner, such as can be applied to a Border.
---------------	--

[types assignable to]	<a href="#">nerRadius</a> <a href="#">x:Object</a>
-----------------------	--

(used by)	<a href="#">Border</a>
-----------	------------------------

[is default constructible]	False
----------------------------	-------

[is nullable]	False
---------------	-------

[text syntax]	<a href="#">nerRadiusSyntax</a>
---------------	---------------------------------

(properties)	
--------------	--

<b>BottomLeft</b>	<a href="#">x:Double</a>
-------------------	--------------------------

(description)	The radius of rounding, in pixels, of the bottom left corner of the object where a <a href="#">nerRadius</a> is applied.
---------------	--

<b>BottomRight</b>	<a href="#">x:Double</a>
--------------------	--------------------------

(description)	The radius of rounding, in pixels, of the bottom right corner of the object where a <a href="#">nerRadius</a> is applied.
---------------	---

<b>TopLeft</b>	<a href="#">x:Double</a>
----------------	--------------------------

(description)	The radius of rounding, in pixels, of the top left corner of the object where a <a href="#">nerRadius</a> is applied.
---------------	---

<b>TopRight</b>	<a href="#">x:Double</a>
-----------------	--------------------------

(description)	The radius of rounding, in pixels, of the top right corner of the object where a <a href="#">nerRadius</a> is applied.
---------------	--

## 1.20 Cursor

(usage)	<a href="#">Arrow</a>   <a href="#">Hand</a>   <a href="#">Wait</a>   <a href="#">IBeam</a>   <a href="#">Stylus</a>   <a href="#">Eraser</a>   <a href="#">SizeNS</a>   <a href="#">SizeWE</a>   <a href="#">None</a>   ...
---------	--

(description)	Represents the image used for the mouse pointer.
---------------	--

[types assignable to]	<a href="#">Cursor</a> <a href="#">x:Object</a>
-----------------------	---

(used by)	<a href="#">Cursors</a> <a href="#">FrameworkElement</a>
-----------	--

[text syntax]	<a href="#">CursorSyntax</a>
---------------	------------------------------

## 1.21 Cursors

(usage)	{x:Static <i>Cursors.StaticPropertyName</i> }
(description)	Defines a set of default mouse pointer images for Silverlight.
[types assignable to]	<a href="#">Cursors x:Object</a>

## 1.22 DependencyObject

(usage)	None.
(description)	Represents an object that participates in the Silverlight property system.
[types assignable to]	<a href="#">DependencyObject x:Object</a>
(used by)	<a href="#">AutomationProperties</a> <a href="#">DesignerProperties</a> <a href="#">InputMethod</a> <a href="#">ScrollViewer</a> <a href="#">ToolTipService</a>

### 1.22.1 AssemblyPart

(usage)	<AssemblyPart />
(description)	An assembly part is an assembly that is to be included in a Silverlight-based application package (.xap).
[types assignable to]	<a href="#">AssemblyPart</a> <a href="#">DependencyObject x:Object</a>
(used by)	<a href="#">AssemblyPartCollection</a> <a href="#">ICollectionOfAssemblyPart</a>
(properties)	
Source	<a href="#">x:String</a>
(description)	The Uri that identifies an assembly as an assembly part.

## 1.22.2 Brush

<b>(usage)</b>	<Brush> string </Brush>
<b>(description)</b>	Defines objects used to paint graphical objects. Types that derive from Brush describe how the area is painted.
<b>[types assignable to]</b>	<a href="#">Brush</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Border</a> <a href="#">Control</a> <a href="#">Glyphs</a> <a href="#">Inline</a> <a href="#">Panel</a> <a href="#">PasswordBox</a> <a href="#">Shape</a> <a href="#">TextBlock</a> <a href="#">TextBox</a> <a href="#">UIElement</a>
<b>[text syntax]</b>	<a href="#">BrushSyntax</a>
<b>(properties)</b>	
<b>Opacity</b>	<a href="#">x:Double</a>
<b>(description)</b>	The degree of opacity of a Brush.
<b>RelativeTransform</b>	<a href="#">Transform</a>
<b>(description)</b>	The transformation that is applied to the brush using relative coordinates.
<b>Transform</b>	<a href="#">Transform</a>
<b>(description)</b>	The transformation that is applied to the brush.

### 1.22.2.1 GradientBrush

<b>(usage)</b>	None.
<b>(description)</b>	A base type that describes a gradient, composed of gradient stops. Types that derive from GradientBrush describe different ways of interpreting gradient stops.
<b>[types assignable to]</b>	<a href="#">GradientBrush</a> <a href="#">Brush</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">GradientStops</a>
<b>(properties)</b>	
<b>ColorInterpolationMode</b>	<a href="#">ColorInterpolationMode</a>
<b>(description)</b>	A ColorInterpolationMode enumeration value that specifies how the gradient's colors are interpolated.
<b>GradientStops</b>	<a href="#">GradientStopCollection</a>
<b>(description)</b>	The brush's gradient stops.
<b>MappingMode</b>	<a href="#">BrushMappingMode</a>
<b>(description)</b>	A BrushMappingMode enumeration value that specifies whether the positioning coordinates of the gradient brush are absolute or relative to the

	output area.
<b>SpreadMethod</b> (description)	<a href="#">GradientSpreadMethod</a> The type of spread method that specifies how to draw a gradient that starts or ends inside the bounds of the object to be painted.

### 1.22.2.1.1 LinearGradientBrush

<b>(usage)</b>	<LinearGradientBrush />
<b>(description)</b>	Paints an area with a linear gradient.
<b>[types assignable to]</b>	<a href="#">LinearGradientBrush</a> <a href="#">GradientBrush</a> <a href="#">Brush</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">GradientStops</a>
<b>(properties)</b>	
<b>EndPoint</b> (description)	<a href="#">Point</a> The ending two-dimensional coordinates of the linear gradient.
<b>StartPoint</b> (description)	<a href="#">Point</a> The starting two-dimensional coordinates of the linear gradient.

### 1.22.2.1.2 RadialGradientBrush

<b>(usage)</b>	<RadialGradientBrush />
<b>(description)</b>	Paints an area with a radial gradient. A focal point defines the beginning of the gradient, and a circle defines the end point of the gradient.
<b>[types assignable to]</b>	<a href="#">RadialGradientBrush</a> <a href="#">GradientBrush</a> <a href="#">Brush</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">GradientStops</a>
<b>(properties)</b>	
<b>Center</b> (description)	<a href="#">Point</a> The center of the outer circle of the radial gradient.
<b>GradientOrigin</b> (description)	<a href="#">Point</a> The location of the focal point that defines the beginning of the gradient.
<b>RadiusX</b>	<a href="#">x:Double</a>



(description)	The horizontal radius of the outer circle of the radial gradient.
RadiusY	<a href="#">x:Double</a>
(description)	The vertical radius of the outer circle of a radial gradient.

## 1.22.2.2 SolidColorBrush

(usage)	<SolidColorBrush />
(description)	Paints an area with a solid color.
[types assignable to]	<a href="#">SolidColorBrush</a> <a href="#">Brush</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Color</a>
(properties)	
Color	<a href="#">Color</a>
(description)	The color of this SolidColorBrush.

## 1.22.2.3 TileBrush

(usage)	None.
(description)	Describes a way to paint a region by using one or more tiles.
[types assignable to]	<a href="#">TileBrush</a> <a href="#">Brush</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(properties)	
AlignmentX	<a href="#">AlignmentX</a>
(description)	The horizontal alignment of content in the TileBrush base tile.
AlignmentY	<a href="#">AlignmentY</a>
(description)	The vertical alignment of content in the TileBrush base tile.
Stretch	<a href="#">Stretch</a>
(description)	A value that specifies how the content of this TileBrush stretches to fit its tiles.

### 1.22.2.3.1 ImageBrush

(usage)	<ImageBrush />
---------	----------------

<b>(description)</b>	Paints an area with an image.
<b>[types assignable to]</b>	<a href="#">ImageBrush</a> <a href="#">TileBrush</a> <a href="#">Brush</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>ImageSource</b>	<a href="#">ImageSource</a>
<b>(description)</b>	The image displayed by this ImageBrush.
<b>(events)</b>	
<b>ImageFailed</b>	Occurs when there is an error associated with image retrieval or format.

### 1.22.2.3.2 VideoBrush

<b>(usage)</b>	<VideoBrush />
<b>(description)</b>	Paints an area with video content.
<b>[types assignable to]</b>	<a href="#">VideoBrush</a> <a href="#">TileBrush</a> <a href="#">Brush</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>SourceName</b>	<a href="#">x:String</a>
<b>(description)</b>	The name of the MediaElement to use as the source of the VideoBrush.

### 1.22.3 ColorKeyFrame

<b>(usage)</b>	None.
<b>(description)</b>	Provides a base type for specific animation key-frame techniques that define an animation segment with a Color target value. Derived types each provide a different key-frame interpolation technique for a Color value that is provided for a ColorAnimationUsingKeyFrames animation.
<b>[types assignable to]</b>	<a href="#">ColorKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">ColorKeyFrameCollection</a> <a href="#">ICollectionOfColorKeyFrame</a>
<b>(properties)</b>	
<b>KeyTime</b>	<a href="#">KeyTime</a>
<b>(description)</b>	The time at which the key frame's target Value should be reached.
<b>Value</b>	<a href="#">Color</a>

(description)	The key frame's target value.
---------------	-------------------------------

### 1.22.3.1 DiscreteColorKeyFrame

(usage)	<DiscreteColorKeyFrame />
---------	---------------------------

(description)	Animates from the Color value of the previous key frame to its own Value using discrete values.
---------------	---

[types assignable to]	<a href="#">DiscreteColorKeyFrame</a> <a href="#">ColorKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
-----------------------	---

### 1.22.3.2 LinearColorKeyFrame

(usage)	<LinearColorKeyFrame />
---------	-------------------------

(description)	Animates from the Color value of the previous key frame to its own Value using linear interpolation.
---------------	--

[types assignable to]	<a href="#">LinearColorKeyFrame</a> <a href="#">ColorKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
-----------------------	---

### 1.22.3.3 SplineColorKeyFrame

(usage)	<SplineColorKeyFrame />
---------	-------------------------

(description)	Animates from the Color value of the previous key frame to its own Value using splined interpolation.
---------------	---

[types assignable to]	<a href="#">SplineColorKeyFrame</a> <a href="#">ColorKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
-----------------------	---

(properties)	
--------------	--

KeySpline	<a href="#">KeySpline</a>
-----------	---------------------------

(description)	The two control points that define animation progress for this key frame.
---------------	---

### 1.22.4 ColumnDefinition

(usage)	<ColumnDefinition />
---------	----------------------

(description)	Defines column-specific properties that apply to Grid objects.
---------------	--

<b>[types assignable to]</b>	<a href="#">ColumnDefinition</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">ColumnDefinitionCollection</a> <a href="#">ICollectionOfColumnDefinition</a>
<b>(properties)</b>	
<b>MaxWidth</b>	<a href="#">x:Double</a>
<b>(description)</b>	A value that represents the maximum width of a ColumnDefinition.
<b>MinWidth</b>	<a href="#">x:Double</a>
<b>(description)</b>	A value that represents the minimum width of a ColumnDefinition.
<b>Width</b>	<a href="#">GridLength</a>
<b>(description)</b>	The calculated width of a ColumnDefinition element, or sets the GridLength value of a column that is defined by the ColumnDefinition.

## 1.22.5 Deployment

<b>(usage)</b>	<Deployment />
<b>(description)</b>	Provides application part and localization information in the application manifest when deploying a Silverlight-based application.
<b>[types assignable to]</b>	<a href="#">Deployment</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Deployment</a>
<b>(properties)</b>	
<b>Parts</b>	<a href="#">AssemblyPartCollection</a>
<b>(description)</b>	A collection of assembly parts that are included in the deployment.
<b>[read only]</b>	True

## 1.22.6 DoubleKeyFrame

<b>(usage)</b>	None.
<b>(description)</b>	A base type that defines an animation segment with its own target value and interpolation technique for a DoubleAnimationUsingKeyFrames.
<b>[types assignable to]</b>	<a href="#">DoubleKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">DoubleKeyFrameCollection</a>

<b>(properties)</b>	
KeyTime	<a href="#">KeyTime</a>
<b>(description)</b>	The time at which the key frame's target Value should be reached.
Value	<a href="#">x:Double</a>
<b>(description)</b>	The key frame's target value.

## 1.22.6.1 DiscreteDoubleKeyFrame

<b>(usage)</b>	<DiscreteDoubleKeyFrame />
<b>(description)</b>	Animates from the Double value of the previous key frame to its own Value using discrete values.
<b>[types assignable to]</b>	<a href="#">DiscreteDoubleKeyFrame</a> <a href="#">DoubleKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>

## 1.22.6.2 LinearDoubleKeyFrame

<b>(usage)</b>	<LinearDoubleKeyFrame />
<b>(description)</b>	Animates from the Double value of the previous key frame to its own Value using linear interpolation.
<b>[types assignable to]</b>	<a href="#">LinearDoubleKeyFrame</a> <a href="#">DoubleKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>

## 1.22.6.3 SplineDoubleKeyFrame

<b>(usage)</b>	<SplineDoubleKeyFrame />
<b>(description)</b>	Animates from the Double value of the previous key frame to its own Value using splined interpolation.
<b>[types assignable to]</b>	<a href="#">SplineDoubleKeyFrame</a> <a href="#">DoubleKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
KeySpline	<a href="#">KeySpline</a>
<b>(description)</b>	The two control points that define animation progress for this key frame.

## 1.22.7 DrawingAttributes

(usage)	<DrawingAttributes />
(description)	Specifies drawing attributes that are used to draw a Stroke.
[types assignable to]	<a href="#">DrawingAttributes</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">Stroke</a>
(properties)	
Color	<a href="#">Color</a>
(description)	The color that is used to draw a Stroke.
Height	<a href="#">x:Double</a>
(description)	The height of the stylus that is used to draw a Stroke.
OutlineColor	<a href="#">Color</a>
(description)	The outline color that is used to draw a Stroke.
Width	<a href="#">x:Double</a>
(description)	The width of the stylus that is used to draw a Stroke.

## 1.22.8 FrameworkTemplate

(usage)	None.
(description)	Creates an element tree of elements.
[types assignable to]	<a href="#">FrameworkTemplate</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Content</a>
[is name scope]	True

### 1.22.8.1 ControlTemplate

(usage)	<ControlTemplate />
(description)	Defines the element tree that is applied as a control template.
[types assignable to]	<a href="#">ControlTemplate</a> <a href="#">FrameworkTemplate</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">Control</a>

[content property]	<a href="#">Content</a>
[is name scope]	True
<b>(properties)</b>	
TargetType	<a href="#">x:XamlType</a>
(description)	The type to which the ControlTemplate is applied.

## 1.22.8.2 DataTemplate

(usage)	<DataTemplate />
(description)	Describes the visual structure of a data object.
[types assignable to]	<a href="#">DataTemplate</a> <a href="#">FrameworkTemplate</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">ComboBox</a> <a href="#">ContentControl</a> <a href="#">ContentPresenter</a> <a href="#">ItemsControl</a>
[content property]	<a href="#">Content</a>
[is name scope]	True

## 1.22.8.3 ItemsPanelTemplate

(usage)	<ItemsPanelTemplate />
(description)	Specifies the panel that the ItemsPresenter creates for the layout of the items of an ItemsControl.
[types assignable to]	<a href="#">ItemsPanelTemplate</a> <a href="#">FrameworkTemplate</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">ItemsControl</a>
[content property]	<a href="#">Content</a>
[is name scope]	True

## 1.22.9 GeneralTransform

(usage)	None.
(description)	Provides generalized transformation support for objects, such as points and rectangles.

[types assignable to]	<a href="#">GeneralTransform</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
-----------------------	--

## 1.22.9.1 Transform

(usage)	<Transform> string </Transform>
---------	---------------------------------

(description)	Defines functionality that enables transformations in a two-dimensional plane.
---------------	--

[types assignable to]	<a href="#">Transform</a> <a href="#">GeneralTransform</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
-----------------------	--

(used by)	<a href="#">Brush</a> <a href="#">Geometry</a> <a href="#">TransformCollection</a> <a href="#">UIElement</a>
-----------	--

[text syntax]	<a href="#">TransformSyntax</a>
---------------	---------------------------------

### 1.22.9.1.1 MatrixTransform

(usage)	<MatrixTransform />
---------	---------------------

(description)	Creates an arbitrary affine matrix transformation that is used to manipulate objects or coordinate systems in a two-dimensional plane.
---------------	--

[types assignable to]	<a href="#">MatrixTransform</a> <a href="#">Transform</a> <a href="#">GeneralTransform</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
-----------------------	--

(properties)	
--------------	--

<b>Matrix</b>	<a href="#">Matrix</a>
---------------	------------------------

(description)	The Matrix that defines this transformation.
---------------	--

### 1.22.9.1.2 RotateTransform

(usage)	<RotateTransform />
---------	---------------------

(description)	Rotates an object clockwise about a specified point in a two-dimensional x-y coordinate system.
---------------	---

[types assignable to]	<a href="#">RotateTransform</a> <a href="#">Transform</a> <a href="#">GeneralTransform</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
-----------------------	--

(properties)	
--------------	--

<b>Angle</b>	<a href="#">x:Double</a>
--------------	--------------------------

(description)	The angle, in degrees, of clockwise rotation.
---------------	---



<b>CenterX</b>	<a href="#">x:Double</a>
(description)	The x-coordinate of the rotation center point.
<b>CenterY</b>	<a href="#">x:Double</a>
(description)	The y-coordinate of the rotation center point.

### 1.22.9.1.3 ScaleTransform

(usage)	<ScaleTransform />
(description)	Scales an object in the two-dimensional x-y coordinate system.
[types assignable to]	<a href="#">ScaleTransform</a> <a href="#">Transform</a> <a href="#">GeneralTransform</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(properties)	
<b>CenterX</b>	<a href="#">x:Double</a>
(description)	The x-coordinate of the center point of this ScaleTransform.
<b>CenterY</b>	<a href="#">x:Double</a>
(description)	The y-coordinate of the center point of this ScaleTransform.
<b>ScaleX</b>	<a href="#">x:Double</a>
(description)	The x-axis scale factor.
<b>ScaleY</b>	<a href="#">x:Double</a>
(description)	The y-axis scale factor.

### 1.22.9.1.4 SkewTransform

(usage)	<SkewTransform />
(description)	Represents a two-dimensional skew.
[types assignable to]	<a href="#">SkewTransform</a> <a href="#">Transform</a> <a href="#">GeneralTransform</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(properties)	
<b>AngleX</b>	<a href="#">x:Double</a>
(description)	The x-axis skew angle, which is measured in degrees counterclockwise from the y-axis.
<b>AngleY</b>	<a href="#">x:Double</a>
(description)	The y-axis skew angle, which is measured in degrees counterclockwise from the

	x-axis.
<b>CenterX</b>	<a href="#">x:Double</a>
<b>(description)</b>	The x-coordinate of the transform center.
<b>CenterY</b>	<a href="#">x:Double</a>
<b>(description)</b>	The y-coordinate of the transform center.

## 1.22.9.1.5 TransformGroup

<b>(usage)</b>	<TransformGroup />
<b>(description)</b>	Represents a composite Transform composed of other Transform objects.
<b>[types assignable to]</b>	<a href="#">TransformGroup</a> <a href="#">Transform</a> <a href="#">GeneralTransform</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Children</a>
<b>(properties)</b>	
<b>Children</b>	<a href="#">TransformCollection</a>
<b>(description)</b>	The collection of child Transform objects.

## 1.22.9.1.6 TranslateTransform

<b>(usage)</b>	<TranslateTransform />
<b>(description)</b>	Translates (moves) an object in the two-dimensional x-y coordinate system.
<b>[types assignable to]</b>	<a href="#">TranslateTransform</a> <a href="#">Transform</a> <a href="#">GeneralTransform</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>X</b>	<a href="#">x:Double</a>
<b>(description)</b>	The distance to translate along the x-axis.
<b>Y</b>	<a href="#">x:Double</a>
<b>(description)</b>	The distance to translate (move) an object along the y-axis.

## 1.22.10 Geometry

<b>(usage)</b>	<code>&lt;Geometry&gt; string &lt;/Geometry&gt;</code>
<b>(description)</b>	Provides a base type for objects that define geometric shapes. Geometry objects can be used for clipping regions and as geometry definitions for rendering two-dimensional graphic data as a Path.
<b>[types assignable to]</b>	<a href="#">Geometry</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Geometry</a> <a href="#">GeometryCollection</a> <a href="#">Path</a> <a href="#">UIElement</a>
<b>[text syntax]</b>	<a href="#">GeometrySyntax</a>
<b>(properties)</b>	
<b>Transform</b>	<a href="#">Transform</a>
<b>(description)</b>	The Transform object applied to a Geometry.

### 1.22.10.1 EllipseGeometry

<b>(usage)</b>	<code>&lt;EllipseGeometry /&gt;</code>
<b>(description)</b>	Represents the geometry of a circle or ellipse.
<b>[types assignable to]</b>	<a href="#">EllipseGeometry</a> <a href="#">Geometry</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>Center</b>	<a href="#">Point</a>
<b>(description)</b>	The center point of the EllipseGeometry.
<b>RadiusX</b>	<a href="#">x:Double</a>
<b>(description)</b>	The x-radius value of the EllipseGeometry.
<b>RadiusY</b>	<a href="#">x:Double</a>
<b>(description)</b>	The y-radius value of the EllipseGeometry.

### 1.22.10.2 GeometryGroup

<b>(usage)</b>	<code>&lt;GeometryGroup /&gt;</code>
<b>(description)</b>	Represents a composite geometry, composed of other Geometry objects.
<b>[types assignable to]</b>	<a href="#">GeometryGroup</a> <a href="#">Geometry</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Children</a>

<b>(properties)</b>	
<b>Children</b>	<a href="#">GeometryCollection</a>
<b>(description)</b>	The GeometryCollection that contains the objects that define this GeometryGroup.
<b>FillRule</b>	<a href="#">FillRule</a>
<b>(description)</b>	How the intersecting areas of the objects contained in this GeometryGroup are combined.

### 1.22.10.3 LineGeometry

<b>(usage)</b>	<LineGeometry />
<b>(description)</b>	Represents the geometry of a line.
<b>[types assignable to]</b>	<a href="#">LineGeometry</a> <a href="#">Geometry</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>EndPoint</b>	<a href="#">Point</a>
<b>(description)</b>	The end point of a line.
<b>StartPoint</b>	<a href="#">Point</a>
<b>(description)</b>	The start point of the line.

### 1.22.10.4 PathGeometry

<b>(usage)</b>	<PathGeometry />
<b>(description)</b>	Represents a complex shape that may be composed of arcs, curves, ellipses, lines, and rectangles.
<b>[types assignable to]</b>	<a href="#">PathGeometry</a> <a href="#">Geometry</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Figures</a>
<b>(properties)</b>	
<b>Figures</b>	<a href="#">PathFigureCollection</a>
<b>(description)</b>	The collection of PathFigure objects that describe the path's contents.
<b>FillRule</b>	<a href="#">FillRule</a>

<b>(description)</b>	A value that determines how the intersecting areas contained in the PathGeometry are combined.
----------------------	--

## 1.22.10.5 RectangleGeometry

<b>(usage)</b>	<RectangleGeometry />
<b>(description)</b>	Describes a two-dimensional rectangular geometry.
<b>[types assignable to]</b>	<a href="#">RectangleGeometry</a> <a href="#">Geometry</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>RadiusX</b>	<a href="#">x:Double</a>
<b>(description)</b>	The x-radius of the ellipse that is used to round the corners of the rectangle.
<b>RadiusY</b>	<a href="#">x:Double</a>
<b>(description)</b>	The y-radius of the ellipse that is used to round the corners of the rectangle.
<b>Rect</b>	<a href="#">Rect</a>
<b>(description)</b>	The dimensions of the rectangle.

## 1.22.11 GradientStop

<b>(usage)</b>	<GradientStop />
<b>(description)</b>	Describes the location and color of a transition point in a gradient.
<b>[types assignable to]</b>	<a href="#">GradientStop</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">GradientStopCollection</a>
<b>[content property]</b>	<a href="#">Color</a>
<b>(properties)</b>	
<b>Color</b>	<a href="#">Color</a>
<b>(description)</b>	The color of the gradient stop.
<b>Offset</b>	<a href="#">x:Double</a>
<b>(description)</b>	The location of the gradient stop within the gradient vector.

## 1.22.12 ImageSource

<b>(usage)</b>	<ImageSource> string </ImageSource>
<b>(description)</b>	Provides an object source type for Source, Source, and ImageSource.
<b>[types assignable to]</b>	<a href="#">ImageSource</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Image</a> <a href="#">ImageBrush</a>
<b>[text syntax]</b>	<a href="#">ImageSourceSyntax</a>

### 1.22.12.1 BitmapImage

<b>(usage)</b>	<BitmapImage />
<b>(description)</b>	Provides the practical object source type for the Source and ImageSource properties.
<b>[types assignable to]</b>	<a href="#">BitmapImage</a> <a href="#">ImageSource</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>UriSource</b>	<a href="#">x:Uri</a>
<b>(description)</b>	The URI of the graphics source file that generated this BitmapImage.
<b>(events)</b>	
<b>DownloadProgress</b>	Occurs when a significant change has occurred in the download progress of the BitmapImage content.

## 1.22.13 Inline

<b>(usage)</b>	None.
<b>(description)</b>	Provides a base for inline flow content element behavior.
<b>[types assignable to]</b>	<a href="#">Inline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">InlineCollection</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>FontFamily</b>	<a href="#">FontFamily</a>

(description)	The preferred top-level font family for the content in this element.
FontSize	<a href="#">x:Double</a>
(description)	The font size for the content in this element.
FontStretch	<a href="#">FontStretch</a>
(description)	The glyph width of the font in a family to select.
FontStyle	<a href="#">FontStyle</a>
(description)	The font style for the content in this element.
FontWeight	<a href="#">FontWeight</a>
(description)	The top-level font weight to select from the font family for the content in this element.
Foreground	<a href="#">Brush</a>
(description)	The Brush to apply to the content in this element.
Language	<a href="#">XmlLanguage</a>
(description)	The language of the content within an Inline element.
TextDecorations	<a href="#">TextDecorationCollection</a>
(description)	The text decorations that are applied to the content within a Inline element.

### 1.22.13.1 LineBreak

(usage)	<LineBreak />
(description)	Represents an inline element that causes a new line to begin in content when rendered in a text container.
[types assignable to]	<a href="#">LineBreak</a> <a href="#">Inline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>
[trim surrounding whitespace]	True

### 1.22.13.2 Run

(usage)	<Run />
(description)	Represents a discrete section of formatted or unformatted text.
[types assignable to]	<a href="#">Run</a> <a href="#">Inline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>

[content property]	<a href="#">Text</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
Text	<a href="#">x:String</a>
(description)	The text contents of the Run.

## 1.22.14 InputMethod

(usage)	None.
(description)	Supports the <a href="#">IsInputMethodEnabled</a> attached property.
[types assignable to]	<a href="#">InputMethod</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(attachable properties)	
InputMethod. IsInputMethodEnabled	<a href="#">x:Boolean</a>
(description)	A value that determines whether input method editors can be used to provide input for the control where the property is attached.
[target type]	<a href="#">DependencyObject</a>

## 1.22.15 KeySpline

(usage)	<code>&lt;KeySpline&gt; string &lt;/KeySpline&gt;</code>
(description)	This type is used by a spline key frame to define animation progress.
[types assignable to]	<a href="#">KeySpline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">SplineColorKeyFrame</a> <a href="#">SplineDoubleKeyFrame</a> <a href="#">SplinePointKeyFrame</a>
[text syntax]	<a href="#">KeySplineSyntax</a>
(properties)	
ControlPoint1	<a href="#">Point</a>
(description)	The first control point used to define a Bezier curve that describes a KeySpline.
ControlPoint2	<a href="#">Point</a>
(description)	The second control point used to define a Bezier curve that describes a



---

 KeySpline.
 

---

## 1.22.16 MultiScaleTileSource

(usage)	None.
(description)	Used to specify the source of Deep Zoom images. The DeepZoomImageTileSource which derives from this type, is used to specify the source of a MultiScaleImage.
[types assignable to]	<a href="#">MultiScaleTileSource</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">MultiScaleImage</a>

---

### 1.22.16.1 DeepZoomImageTileSource

(usage)	<DeepZoomImageTileSource> string </DeepZoomImageTileSource>
(description)	Used to specify the source of a MultiScaleImage.
[types assignable to]	<a href="#">DeepZoomImageTileSource</a> <a href="#">MultiScaleTileSource</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[text syntax]	<a href="#">DeepZoomImageTileSourceSyntax</a>
(properties)	
UriSource	<a href="#">x:Uri</a>
(description)	The source Uri of the DeepZoomImageTileSource.

---

## 1.22.17 ObjectKeyFrame

(usage)	None.
(description)	Defines an animation segment with its own target value and interpolation technique for an ObjectAnimationUsingKeyFrames.
[types assignable to]	<a href="#">ObjectKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">ObjectKeyFrameCollection</a>
(properties)	
KeyTime	<a href="#">KeyTime</a>
(description)	The time at which the key frame's target Value should be reached.

---

<b>Value</b>	<a href="#">x:Object</a>
<b>(description)</b>	The key frame's target value.

## 1.22.17.1 DiscreteObjectKeyFrame

<b>(usage)</b>	<code>&lt;DiscreteObjectKeyFrame /&gt;</code>
<b>(description)</b>	Animates from the Object value of the previous key frame to its own Value using discrete values.
<b>[types assignable to]</b>	<a href="#">DiscreteObjectKeyFrame</a> <a href="#">ObjectKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>

## 1.22.18 PathFigure

<b>(usage)</b>	<code>&lt;PathFigure /&gt;</code>
<b>(description)</b>	Represents a subsection of a geometry, a single connected series of two-dimensional geometric segments.
<b>[types assignable to]</b>	<a href="#">PathFigure</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">PathFigureCollection</a>
<b>[content property]</b>	<a href="#">Segments</a>
<b>(properties)</b>	
<b>IsClosed</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether this figure's first and last segments are connected.
<b>IsFilled</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether the contained area of this PathFigure is to be used for hit-testing, rendering, and clipping.
<b>Segments</b>	<a href="#">PathSegmentCollection</a>
<b>(description)</b>	The collection of segments that define the shape of this PathFigure object.
<b>StartPoint</b>	<a href="#">Point</a>
<b>(description)</b>	The Point where the PathFigure begins.

## 1.22.19 PathSegment

(usage)	None.
(description)	Represents a segment of a PathFigure object.
[types assignable to]	<a href="#">PathSegment</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">PathSegmentCollection</a>

### 1.22.19.1 ArcSegment

(usage)	<ArcSegment />
(description)	Represents an elliptical arc between two points.
[types assignable to]	<a href="#">ArcSegment</a> <a href="#">PathSegment</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(properties)	
IsLargeArc	<a href="#">x:Boolean</a>
(description)	A value that indicates whether the arc should be greater than 180 degrees.
Point	<a href="#">Point</a>
(description)	The endpoint of the elliptical arc.
RotationAngle	<a href="#">x:Double</a>
(description)	The amount (in degrees) by which the ellipse is rotated about the x-axis.
Size	<a href="#">Size</a>
(description)	The x- and y-radius of the arc as a Size structure.
SweepDirection	<a href="#">SweepDirection</a>
(description)	A value that specifies whether the arc is drawn in the Clockwise or Counterclockwise direction.

### 1.22.19.2 BezierSegment

(usage)	<BezierSegment />
(description)	Represents a cubic Bezier curve drawn between two points.
[types assignable to]	<a href="#">BezierSegment</a> <a href="#">PathSegment</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(properties)	

<b>Point1</b>	<a href="#">Point</a>
(description)	The first control point of the curve.
<b>Point2</b>	<a href="#">Point</a>
(description)	The second control point of the curve.
<b>Point3</b>	<a href="#">Point</a>
(description)	The end point of the curve.

### 1.22.19.3 LineSegment

(usage)	<LineSegment />
(description)	Represents a line drawn between two points, which can be part of a PathFigure within Path data.
[types assignable to]	<a href="#">LineSegment</a> <a href="#">PathSegment</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(properties)	
<b>Point</b>	<a href="#">Point</a>
(description)	The end point of the line segment.

### 1.22.19.4 PolyBezierSegment

(usage)	<PolyBezierSegment />
(description)	Represents one or more cubic Bezier curves.
[types assignable to]	<a href="#">PolyBezierSegment</a> <a href="#">PathSegment</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Points</a>
(properties)	
<b>Points</b>	<a href="#">PointCollection</a>
(description)	The PointCollection that defines this PolyBezierSegment object.

### 1.22.19.5 PolyLineSegment

[MS-SLXV] – v0.9

Silverlight Xaml Vocabulary Specification 2008

Copyright © 2008 Microsoft Corporation.

Release: October 2008

<b>(usage)</b>	<PolyLineSegment />
<b>(description)</b>	Represents a set of line segments defined by a PointCollection with each Point specifying the end point of a line segment.
<b>[types assignable to]</b>	<a href="#">PolyLineSegment</a> <a href="#">PathSegment</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Points</a>
<b>(properties)</b>	
<b>Points</b>	<a href="#">PointCollection</a>
<b>(description)</b>	The collection of Point values that defines this PolyLineSegment object.

## 1.22.19.6 PolyQuadraticBezierSegment

<b>(usage)</b>	<PolyQuadraticBezierSegment />
<b>(description)</b>	Represents a set of quadratic Bezier segments.
<b>[types assignable to]</b>	<a href="#">PolyQuadraticBezierSegment</a> <a href="#">PathSegment</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Points</a>
<b>(properties)</b>	
<b>Points</b>	<a href="#">PointCollection</a>
<b>(description)</b>	The PointCollection that defines this PolyQuadraticBezierSegment object.

## 1.22.19.7 QuadraticBezierSegment

<b>(usage)</b>	<QuadraticBezierSegment />
<b>(description)</b>	Creates a quadratic Bezier curve between two points in a PathFigure.
<b>[types assignable to]</b>	<a href="#">QuadraticBezierSegment</a> <a href="#">PathSegment</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>Point1</b>	<a href="#">Point</a>
<b>(description)</b>	The control point of the curve.
<b>Point2</b>	<a href="#">Point</a>
<b>(description)</b>	The end Point of this QuadraticBezierSegment.

## 1.22.20 PointKeyFrame

<b>(usage)</b>	None.
<b>(description)</b>	Defines an animation segment with its own target value and interpolation technique for a <a href="#">PointAnimationUsingKeyFrames</a> .
<b>[types assignable to]</b>	<a href="#">PointKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">PointKeyFrameCollection</a>
<b>(properties)</b>	
<b>KeyTime</b>	<a href="#">KeyTime</a>
<b>(description)</b>	The time at which the key frame's target Value should be reached.
<b>Value</b>	<a href="#">Point</a>
<b>(description)</b>	The key frame's target value.

### 1.22.20.1 DiscretePointKeyFrame

<b>(usage)</b>	<DiscretePointKeyFrame />
<b>(description)</b>	Animates from the <a href="#">Point</a> value of the previous key frame to its own <a href="#">Value</a> using discrete frames.
<b>[types assignable to]</b>	<a href="#">DiscretePointKeyFrame</a> <a href="#">PointKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>

### 1.22.20.2 LinearPointKeyFrame

<b>(usage)</b>	<LinearPointKeyFrame />
<b>(description)</b>	Animates from the <a href="#">Point</a> value of the previous key frame to its own <a href="#">Value</a> using linear interpolation.
<b>[types assignable to]</b>	<a href="#">LinearPointKeyFrame</a> <a href="#">PointKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>

### 1.22.20.3 SplinePointKeyFrame

<b>(usage)</b>	<SplinePointKeyFrame />
----------------	-------------------------

(description)	Animates from the Point value of the previous key frame to its own Value using splined interpolation.
[types assignable to]	<a href="#">SplinePointKeyFrame</a> <a href="#">PointKeyFrame</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(properties)	
KeySpline	<a href="#">KeySpline</a>
(description)	The two control points that define animation progress for this key frame.

## 1.22.21 PresentationFrameworkCollectionOfT

(usage)	None.
(description)	Provides a common collection type for Silverlight collections.
[types assignable to]	<a href="#">PresentationFrameworkCollectionOfT</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[is list]	True
[allowed types]	<a href="#">I</a>

## 1.22.22 ResourceDictionary

(usage)	<ResourceDictionary />
(description)	Provides a dictionary that contains keyed resources used by components of a Silverlight-based application.
[types assignable to]	<a href="#">ResourceDictionary</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[is dictionary]	True

## 1.22.23 RowDefinition

(usage)	<RowDefinition />
(description)	Defines row-specific properties that apply to Grid elements.
[types assignable to]	<a href="#">RowDefinition</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">RowDefinitionCollection</a>
(properties)	

<b>Height</b>	<a href="#">GridLength</a>
(description)	The calculated height of a RowDefinition element, or sets the GridLength value of a row that is defined by the RowDefinition.
<b>MaxHeight</b>	<a href="#">x:Double</a>
(description)	A value that represents the maximum height of a RowDefinition.
<b>MinHeight</b>	<a href="#">x:Double</a>
(description)	A value that represents the minimum allowed height of a RowDefinition.

## 1.22.24 SetterBase

(usage)	None.
(description)	Represents the base type for value setters.
[types assignable to]	<a href="#">SetterBase</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">SetterBaseCollection</a>

### 1.22.24.1 Setter

(usage)	<code>&lt;Setter /&gt;</code>
(description)	Applies a value to a property in a Style.
[types assignable to]	<a href="#">Setter</a> <a href="#">SetterBase</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(properties)	
<b>Property</b>	<a href="#">DependencyProperty</a>
(description)	The property to apply the Value to.
<b>Value</b>	<a href="#">x:Object</a>
(description)	The value to apply to the property that is specified by the Setter.

## 1.22.25 Stroke

(usage)	<code>&lt;Stroke /&gt;</code>
---------	-------------------------------



<b>(description)</b>	Represents a collection of points that correspond to a stylus-down, move, and stylus-up sequence.
<b>[types assignable to]</b>	<a href="#">Stroke</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">StrokeCollection</a>
<b>(properties)</b>	
<b>DrawingAttributes</b>	<a href="#">DrawingAttributes</a>
<b>(description)</b>	The properties of the stroke, such as Height, Width, Color, or OutlineColor.
<b>StylusPoints</b>	<a href="#">StylusPointCollection</a>
<b>(description)</b>	The stylus points of the Stroke.

## 1.22.26 Style

<b>(usage)</b>	<Style />
<b>(description)</b>	Contains property setters that can be shared between instances of a type.
<b>[types assignable to]</b>	<a href="#">Style</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">ComboBox</a> <a href="#">FrameworkElement</a> <a href="#">ListBox</a>
<b>[content property]</b>	<a href="#">Setters</a>
<b>[dictionary key property]</b>	<a href="#">TargetType</a>
<b>(properties)</b>	
<b>Setters</b>	<a href="#">SetterBaseCollection</a>
<b>(description)</b>	A collection of Setter objects.
<b>[read only]</b>	True
<b>TargetType</b>	<a href="#">x:XamlType</a>
<b>(description)</b>	The type for which the style is intended.

## 1.22.27 Timeline

<b>(usage)</b>	None.
<b>(description)</b>	Defines a segment of time.
<b>[types assignable to]</b>	<a href="#">Timeline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Storyboard</a> <a href="#">TimelineCollection</a>

<b>(properties)</b>	
<b>AutoReverse</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether the timeline plays in reverse after it completes a forward iteration.
<b>BeginTime</b>	<a href="#">NullableOfTimeSpan</a>
<b>(description)</b>	The time at which this Timeline should begin.
<b>Duration</b>	<a href="#">Duration</a>
<b>(description)</b>	The length of time for which this timeline plays, not counting repetitions.
<b>FillBehavior</b>	<a href="#">FillBehavior</a>
<b>(description)</b>	A value that specifies how the animation behaves after it reaches the end of its active period.
<b>RepeatBehavior</b>	<a href="#">RepeatBehavior</a>
<b>(description)</b>	The repeating behavior of this timeline.
<b>SpeedRatio</b>	<a href="#">x:Double</a>
<b>(description)</b>	The rate, relative to its parent, at which time progresses for this Timeline.
<b>(events)</b>	
<b>Completed</b>	Occurs when the Storyboard object has completed playing.

## 1.22.27.1 ColorAnimation

<b>(usage)</b>	<ColorAnimation />
<b>(description)</b>	Animates the value of a <code>Color</code> property between two target values using linear interpolation over a specified <code>Duration</code> .
<b>[types assignable to]</b>	<a href="#">ColorAnimation</a> <a href="#">Timeline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>By</b>	<a href="#">NullableOfColor</a>
<b>(description)</b>	The total amount by which the animation changes its starting value.
<b>From</b>	<a href="#">NullableOfColor</a>
<b>(description)</b>	The animation's starting value.
<b>To</b>	<a href="#">NullableOfColor</a>
<b>(description)</b>	The animation's ending value.

## 1.22.27.2 ColorAnimationUsingKeyFrames

<b>(usage)</b>	<ColorAnimationUsingKeyFrames />
<b>(description)</b>	Animates the value of a Color property along a set of KeyFrames over a specified Duration.
<b>[types assignable to]</b>	<a href="#">ColorAnimationUsingKeyFrames</a> <a href="#">Timeline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">KeyFrames</a>
<b>(properties)</b>	
<b>KeyFrames</b>	<a href="#">ColorKeyFrameCollection</a>
<b>(description)</b>	The collection of ColorKeyFrame objects that define the animation.
<b>[read only]</b>	True

## 1.22.27.3 DoubleAnimation

<b>(usage)</b>	<DoubleAnimation />
<b>(description)</b>	Animates the value of a Double property between two target values using linear interpolation over a specified Duration.
<b>[types assignable to]</b>	<a href="#">DoubleAnimation</a> <a href="#">Timeline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>By</b>	<a href="#">NullableOfDouble</a>
<b>(description)</b>	The total amount by which the animation changes its starting value.
<b>From</b>	<a href="#">NullableOfDouble</a>
<b>(description)</b>	The animation's starting value.
<b>To</b>	<a href="#">NullableOfDouble</a>
<b>(description)</b>	The animation's ending value.

## 1.22.27.4 DoubleAnimationUsingKeyFrames

<b>(usage)</b>	<DoubleAnimationUsingKeyFrames />
----------------	-----------------------------------

<b>(description)</b>	Animates the value of a Double property along a set of KeyFrames.
<b>[types assignable to]</b>	<a href="#">DoubleAnimationUsingKeyFrames</a> <a href="#">Timeline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">KeyFrames</a>
<b>(properties)</b>	
<b>KeyFrames</b>	<a href="#">DoubleKeyFrameCollection</a>
<b>(description)</b>	The collection of DoubleKeyFrame objects that define the animation.
<b>[read only]</b>	True

## 1.22.27.5 ObjectAnimationUsingKeyFrames

<b>(usage)</b>	<ObjectAnimationUsingKeyFrames />
<b>(description)</b>	Animates the value of an Object property along a set of KeyFrames over a specified Duration.
<b>[types assignable to]</b>	<a href="#">ObjectAnimationUsingKeyFrames</a> <a href="#">Timeline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">KeyFrames</a>
<b>(properties)</b>	
<b>KeyFrames</b>	<a href="#">ObjectKeyFrameCollection</a>
<b>(description)</b>	The collection of ObjectKeyFrame objects that define the animation.
<b>[read only]</b>	True

## 1.22.27.6 PointAnimation

<b>(usage)</b>	<PointAnimation />
<b>(description)</b>	Animates the value of a Point property between two target values using linear interpolation over a specified Duration.
<b>[types assignable to]</b>	<a href="#">PointAnimation</a> <a href="#">Timeline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(properties)</b>	
<b>By</b>	<a href="#">NullableOfPoint</a>
<b>(description)</b>	The total amount by which the animation changes its starting value.
<b>From</b>	<a href="#">NullableOfPoint</a>

(description)	The animation's starting value.
To	<a href="#">NullableOfPoint</a>
(description)	The animation's ending value.

## 1.22.27.7 PointAnimationUsingKeyFrames

(usage)	<PointAnimationUsingKeyFrames />
(description)	Animates the value of a Point property along a set of KeyFrames.
[types assignable to]	<a href="#">PointAnimationUsingKeyFrames</a> <a href="#">Timeline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">KeyFrames</a>
(properties)	
KeyFrames	<a href="#">PointKeyFrameCollection</a>
(description)	The collection of PointKeyFrame objects that define the animation.
[read only]	True

## 1.22.27.8 Storyboard

(usage)	<Storyboard />
(description)	Controls animations with a timeline, and provides object and property targeting information for its child animations.
[types assignable to]	<a href="#">Storyboard</a> <a href="#">Timeline</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">BeginStoryboard</a> <a href="#">VisualState</a>
[content property]	<a href="#">Children</a>
(properties)	
Children	<a href="#">TimelineCollection</a>
(description)	The collection of child Timeline objects.
[read only]	True
(attachable properties)	
Storyboard.TargetName	<a href="#">x:String</a>
(description)	The name of the object to animate.
[target type]	<a href="#">Timeline</a>

<b>Storyboard.TargetProperty</b>	<a href="#">PropertyPath</a>
(description)	The name of the property that should be animated.
[target type]	<a href="#">Timeline</a>

## 1.22.28 TimelineMarker

(usage)	<TimelineMarker />
(description)	Represents metadata associated with a specific point in a media file.
[types assignable to]	<a href="#">TimelineMarker</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">TimelineMarkerCollection</a>
(properties)	
Text	<a href="#">x:String</a>
(description)	The text value of a TimelineMarker.
Time	<a href="#">x:TimeSpan</a>
(description)	The time at which a TimelineMarker is reached.
Type	<a href="#">x:String</a>
(description)	The marker type of a TimelineMarker.

## 1.22.29 TriggerAction

(usage)	None.
(description)	Serves as the base type for BeginStoryboard.
[types assignable to]	<a href="#">TriggerAction</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">TriggerActionCollection</a>

### 1.22.29.1 BeginStoryboard

(usage)	<BeginStoryboard />
---------	---------------------

<b>(description)</b>	A trigger action that begins a Storyboard and distributes its animations to their targeted objects and properties.
<b>[types assignable to]</b>	<a href="#">BeginInitStoryboard</a> <a href="#">TriggerAction</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Storyboard</a>
<b>(properties)</b>	
<b>Storyboard</b>	<a href="#">Storyboard</a>
<b>(description)</b>	The Storyboard that this BeginStoryboard starts.

## 1.22.30 TriggerBase

<b>(usage)</b>	None.
<b>(description)</b>	Serves as the base type for EventTrigger.
<b>[types assignable to]</b>	<a href="#">TriggerBase</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">TriggerCollection</a>

### 1.22.30.1 EventTrigger

<b>(usage)</b>	<EventTrigger />
<b>(description)</b>	Represents a trigger that applies a set of actions (animation storyboards) in response to an event.
<b>[types assignable to]</b>	<a href="#">EventTrigger</a> <a href="#">TriggerBase</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Actions</a>
<b>(properties)</b>	
<b>Actions</b>	<a href="#">TriggerActionCollection</a>
<b>(description)</b>	The collection of BeginStoryboard objects that this EventTrigger maintains.
<b>[read only]</b>	True
<b>RoutedEvent</b>	<a href="#">RoutedEvent</a>
<b>(description)</b>	The name of the event that initiates the trigger.

## 1.22.31 UIElement

<b>(usage)</b>	None.
<b>(description)</b>	UIElement is a base type for most of the objects that have visual appearance and can process basic input in Silverlight.
<b>[types assignable to]</b>	<a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Application</a> <a href="#">AutomationProperties</a> <a href="#">Border</a> <a href="#">Canvas</a> <a href="#">Popup</a> <a href="#">UIElementCollection</a>
<b>(properties)</b>	
<b>Clip</b>	<a href="#">Geometry</a>
<b>(description)</b>	The Geometry used to define the outline of the contents of a UIElement.
<b>IsHitTestVisible</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	Whether the contained area of this UIElement can return true values for hit testing.
<b>Opacity</b>	<a href="#">x:Double</a>
<b>(description)</b>	The degree of the object's opacity.
<b>OpacityMask</b>	<a href="#">Brush</a>
<b>(description)</b>	The brush used to alter the opacity of regions of this object.
<b>RenderTransform</b>	<a href="#">Transform</a>
<b>(description)</b>	Transform information that affects the rendering position of a UIElement.
<b>RenderTransformOrigin</b>	<a href="#">Point</a>
<b>(description)</b>	The origin point of any possible render transform declared by RenderTransform, relative to the bounds of the UIElement.
<b>UseLayoutRounding</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that determines whether rendering for the object and its subtree should use rounding behavior that aligns rendering to whole pixels.
<b>Visibility</b>	<a href="#">Visibility</a>
<b>(description)</b>	The visibility of a UIElement. A UIElement that is not visible does not render and does not communicate its desired size to layout.
<b>(events)</b>	
<b>GotFocus</b>	Occurs when a UIElement receives focus.
<b>KeyDown</b>	Occurs when a keyboard key is pressed while the UIElement has focus.
<b>KeyUp</b>	Occurs when a keyboard key is released while the UIElement has focus.
<b>LostFocus</b>	Occurs when a UIElement loses focus.



<b>LostMouseCapture</b>	Occurs when the object loses mouse (or stylus) capture.
<b>MouseEnter</b>	Occurs when the mouse (or a stylus) enters the bounding area of a UIElement.
<b>MouseLeave</b>	Occurs when the mouse (or the stylus) leaves the bounding area of a UIElement.
<b>MouseLeftButtonDown</b>	Occurs when the left mouse button is pressed (or when the tip of the stylus touches the tablet PC) while the mouse pointer is over a UIElement.
<b>MouseLeftButtonUp</b>	Occurs when the left mouse button is released (or the tip of the stylus is removed from the tablet PC) while the mouse (or the stylus) is over a UIElement (or while a UIElement holds mouse capture).
<b>MouseMove</b>	Occurs when the coordinate position of the mouse (or stylus) changes while over a UIElement (or while a UIElement holds mouse capture).

## 1.22.31.1 FrameworkElement

<b>(usage)</b>	None.
<b>(description)</b>	Provides a framework of common APIs for objects that participate in Silverlight layout. Also defines APIs related to data binding, object tree, and object lifetime feature areas in Silverlight.
<b>[types assignable to]</b>	<a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Grid</a> <a href="#">VisualStateManager</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>Cursor</b>	<a href="#">Cursor</a>
<b>(description)</b>	The cursor that displays while the mouse pointer is over a FrameworkElement.
<b>DataContext</b>	<a href="#">x:Object</a>
<b>(description)</b>	The data context for a FrameworkElement when it participates in data binding.
<b>Height</b>	<a href="#">x:Double</a>
<b>(description)</b>	The suggested height of a FrameworkElement.
<b>HorizontalAlignment</b>	<a href="#">HorizontalAlignment</a>
<b>(description)</b>	The horizontal alignment characteristics applied to a FrameworkElement when it is composed within a layout parent, such as a panel or items control.
<b>Language</b>	<a href="#">XmlLanguage</a>
<b>(description)</b>	Localization/globalization language information that applies to a FrameworkElement.

<b>Margin</b>	<a href="#">Thickness</a>
(description)	The outer margin of a FrameworkElement.
<b>MaxHeight</b>	<a href="#">x:Double</a>
(description)	The maximum height constraint of a FrameworkElement.
<b>MaxWidth</b>	<a href="#">x:Double</a>
(description)	The maximum width constraint of a FrameworkElement.
<b>MinHeight</b>	<a href="#">x:Double</a>
(description)	The minimum height constraint of a FrameworkElement.
<b>MinWidth</b>	<a href="#">x:Double</a>
(description)	The minimum width constraint of a FrameworkElement.
<b>Style</b>	<a href="#">Style</a>
(description)	An instance Style that is applied for this object during rendering.
<b>Tag</b>	<a href="#">x:Object</a>
(description)	An arbitrary object value that can be used to store custom information about this object.
<b>Triggers</b>	<a href="#">TriggerCollection</a>
(description)	The collection of triggers for animations that are defined for a FrameworkElement.
[read only]	True
<b>VerticalAlignment</b>	<a href="#">VerticalAlignment</a>
(description)	The vertical alignment characteristics applied to a FrameworkElement when it is composed within a parent object such as a panel or items control.
<b>Width</b>	<a href="#">x:Double</a>
(description)	The width of a FrameworkElement.
<b>(events)</b>	
<b>BindingValidationError</b>	Occurs when a data validation error is reported by a binding source.
<b>LayoutUpdated</b>	Occurs when the layout of the various child objects associated with the current Silverlight plug-in instance changes.
<b>Loaded</b>	Occurs when a FrameworkElement has completed layout passes, has rendered, and is ready for interaction.
<b>SizeChanged</b>	Occurs when either the ActualHeight or the ActualWidth properties change value on a FrameworkElement.

## 1.22.31.1.1 Border

(usage)	<code>&lt;Border /&gt;</code>
(description)	Draws a border, background, or both around another object.
[types assignable to]	<a href="#">Border</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Child</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
Background	<a href="#">Brush</a>
(description)	The Brush that fills the background of the border.
BorderBrush	<a href="#">Brush</a>
(description)	The Brush that is used to create the border.
BorderThickness	<a href="#">Thickness</a>
(description)	The thickness of the border.
Child	<a href="#">UIElement</a>
(description)	The child element to draw the border around.
CornerRadius	<a href="#">CornerRadius</a>
(description)	The radius for the corners of the border.
Padding	<a href="#">Thickness</a>
(description)	The distance between the border and its child object.

## 1.22.31.1.2 ContentPresenter

(usage)	<code>&lt;ContentPresenter /&gt;</code>
(description)	Displays the content of a ContentPresenter.
[types assignable to]	<a href="#">ContentPresenter</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Content</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
Content	<a href="#">x:Object</a>
(description)	The data that is used to generate the child elements of a ContentPresenter.

<b>ContentTemplate</b>	<a href="#">DataTemplate</a>
(description)	The template that is used to display the content of the control.

## 1.22.31.1.2.1 ScrollContentPresenter

(usage)	<ScrollContentPresenter />
(description)	Displays the content of a ScrollViewer control.
[types assignable to]	<a href="#">ScrollContentPresenter</a> <a href="#">ContentPresenter</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Content</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
<b>CanHorizontallyScroll</b>	<a href="#">x:Boolean</a>
(description)	A value that indicates whether scrolling on the horizontal axis is possible.
<b>CanVerticallyScroll</b>	<a href="#">x:Boolean</a>
(description)	A value that indicates whether scrolling on the vertical axis is possible.
<b>ScrollOwner</b>	<a href="#">ScrollViewer</a>
(description)	The ScrollViewer element that controls scrolling behavior.

## 1.22.31.1.3 Control

(usage)	None.
(description)	Represents the base type for UI elements that use a ControlTemplate to define their appearance.
[types assignable to]	<a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
<b>Background</b>	<a href="#">Brush</a>
(description)	A brush that provides the background of the control.
<b>BorderBrush</b>	<a href="#">Brush</a>

(description)	A brush that describes the border background of a control.
<b>BorderThickness</b>	<a href="#">Thickness</a>
(description)	The border thickness of a control.
<b>FontFamily</b>	<a href="#">FontFamily</a>
(description)	The font used to display text in the control.
<b>FontSize</b>	<a href="#">x:Double</a>
(description)	The size of the text in this control.
<b>FontStretch</b>	<a href="#">FontStretch</a>
(description)	The degree to which a font is condensed or expanded on the screen.
<b>FontStyle</b>	<a href="#">FontStyle</a>
(description)	The style in which the text is rendered.
<b>FontWeight</b>	<a href="#">FontWeight</a>
(description)	The thickness of the specified font.
<b>Foreground</b>	<a href="#">Brush</a>
(description)	A brush that describes the foreground color.
<b>HorizontalContentAlignment</b>	<a href="#">HorizontalAlignment</a>
(description)	The horizontal alignment of the control's content.
<b>IsEnabled</b>	<a href="#">x:Boolean</a>
(description)	A value indicating whether the user can interact with the control.
<b>IsTabStop</b>	<a href="#">x:Boolean</a>
(description)	A value that indicates whether a control is included in tab navigation.
<b>Padding</b>	<a href="#">Thickness</a>
(description)	The padding inside a control.
<b>TabIndex</b>	<a href="#">x:Int32</a>
(description)	A value that determines the order in which elements receive focus when the user navigates through controls by using the TAB key.
<b>TabNavigation</b>	<a href="#">KeyboardNavigationMode</a>
(description)	A value that modifies how tabbing and TabIndex work for this control.
<b>Template</b>	<a href="#">ControlTemplate</a>
(description)	A control template.
<b>VerticalContentAlignment</b>	<a href="#">VerticalAlignment</a>
(description)	The vertical alignment of the control's content.
<b>(events)</b>	

**IsEnabledChanged**Occurs when the `IsEnabled` property changes.

## 1.22.31.1.3.1 ContentControl

**(usage)**

&lt;ContentControl /&gt;

**(description)**Represents a control with a single piece of content. Controls such as `Button`, `CheckBox`, and `ScrollViewer` directly or indirectly inherit from this type.**[types assignable to]**[ContentControl](#) [Control](#) [FrameworkElement](#) [UIElement](#) [DependencyObject](#)  
[x:Object](#)**[content property]**[Content](#)**[xml lang property]**[Language](#)**(properties)****Content**[x:Object](#)**(description)**The value of the `ContentControl` property.**ContentTemplate**[DataTemplate](#)**(description)**The data template that is used to display the content of the `ContentControl`.

### 1.22.31.1.3.1.1 ButtonBase

**(usage)**

None.

**(description)**Represents the base type for all button controls, such as `Button`, `RepeatButton`, and `HyperlinkButton`.**[types assignable to]**[ButtonBase](#) [ContentControl](#) [Control](#) [FrameworkElement](#) [UIElement](#)  
[DependencyObject](#) [x:Object](#)**[content property]**[Content](#)**[xml lang property]**[Language](#)**(properties)****ClickMode**[ClickMode](#)**(description)**When the `Click` event occurs.**IsFocused**[x:Boolean](#)**(description)**

A value that determines whether the button has focus.

[MS-SLXV] – v0.9

Silverlight Xaml Vocabulary Specification 2008

Copyright © 2008 Microsoft Corporation.

Release: October 2008

<b>IsMouseOver</b>	<a href="#">x:Boolean</a>
(description)	A value indicating whether the mouse pointer is located over this button control.
<b>IsPressed</b>	<a href="#">x:Boolean</a>
(description)	A value that indicates whether a ButtonBase is currently in a pressed state.
<b>(events)</b>	
<b>Click</b>	Occurs when a Button is clicked.

## 1.22.31.1.3.1.1.1 Button

<b>(usage)</b>	<code>&lt;Button /&gt;</code>
<b>(description)</b>	Represents a button control, which responds to the Click event.
<b>[types assignable to]</b>	<a href="#">Button</a> <a href="#">ButtonBase</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>

## 1.22.31.1.3.1.1.2 HyperlinkButton

<b>(usage)</b>	<code>&lt;HyperlinkButton /&gt;</code>
<b>(description)</b>	Represents a button control that displays a hyperlink.
<b>[types assignable to]</b>	<a href="#">HyperlinkButton</a> <a href="#">ButtonBase</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>NavigateUri</b>	<a href="#">x:Uri</a>
(description)	The URI to navigate to when the HyperlinkButton is clicked.
<b>TargetName</b>	<a href="#">x:String</a>
(description)	The name of a target window or frame to navigate to within the Web page specified by the NavigateUri property.

### 1.22.31.1.3.1.1.3 RepeatButton

<b>(usage)</b>	<RepeatButton />
<b>(description)</b>	Represents a control that raises its Click event repeatedly from the time it is pressed until it is released.
<b>[types assignable to]</b>	<a href="#">RepeatButton</a> <a href="#">ButtonBase</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>Delay</b>	<a href="#">x:Int32</a>
<b>(description)</b>	The time, in milliseconds, the RepeatButton waits when it is pressed before it starts repeating the click action.
<b>Interval</b>	<a href="#">x:Int32</a>
<b>(description)</b>	The time, in milliseconds, between repetitions of the click action, as soon as repeating starts.

### 1.22.31.1.3.1.1.4 ToggleButton

<b>(usage)</b>	<ToggleButton />
<b>(description)</b>	Base type for controls that can switch states, such as CheckBox and RadioButton.
<b>[types assignable to]</b>	<a href="#">ToggleButton</a> <a href="#">ButtonBase</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>IsChecked</b>	<a href="#">NullableOfBoolean</a>
<b>(description)</b>	Whether the ToggleButton is checked.
<b>IsThreeState</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	Whether the control supports two or three states.



<b>(events)</b>	
<b>Checked</b>	Occurs when a ToggleButton is checked.
<b>Indeterminate</b>	Occurs when the state of a ToggleButton is switched to the indeterminate state.
<b>Unchecked</b>	Occurs when a ToggleButton is unchecked.

### 1.22.31.1.3.1.1.4.1 CheckBox

<b>(usage)</b>	<CheckBox />
<b>(description)</b>	Represents a control that a user can select (check) or clear (uncheck).
<b>[types assignable to]</b>	<a href="#">CheckBox</a> <a href="#">ToggleButton</a> <a href="#">ButtonBase</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>

### 1.22.31.1.3.1.1.4.2 RadioButton

<b>(usage)</b>	<RadioButton />
<b>(description)</b>	Represents a button that allows a user to select a single option from a group of options.
<b>[types assignable to]</b>	<a href="#">RadioButton</a> <a href="#">ToggleButton</a> <a href="#">ButtonBase</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>GroupName</b>	<a href="#">x:String</a>
<b>(description)</b>	The name that specifies which RadioButton controls are mutually exclusive.

### 1.22.31.1.3.1.2 ListBoxItem

<b>(usage)</b>	<ListBoxItem />
----------------	-----------------

<b>(description)</b>	Represents a selectable item in a ListBox.
<b>[types assignable to]</b>	<a href="#">ListBoxItem</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>IsSelected</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether a ListBoxItem is selected.

### 1.22.31.1.3.1.2.1 ComboBoxItem

<b>(usage)</b>	<ComboBoxItem />
<b>(description)</b>	Represents a selectable item contained in a ComboBox control.
<b>[types assignable to]</b>	<a href="#">ComboBoxItem</a> <a href="#">ListBoxItem</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>

### 1.22.31.1.3.1.3 ScrollViewer

<b>(usage)</b>	<ScrollViewer />
<b>(description)</b>	Represents a scrollable area that can contain other visible elements.
<b>[types assignable to]</b>	<a href="#">ScrollViewer</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">ScrollContentPresenter</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>ComputedHorizontalScrollBarVisibility</b>	<a href="#">Visibility</a>
<b>(description)</b>	A value that indicates whether the horizontal ScrollBar is visible.

<b>ComputedVerticalScrollBarVisibility</b>	<a href="#">Visibility</a>
(description)	A value that indicates whether the vertical ScrollBar is visible.
<b>ExtentHeight</b>	<a href="#">x:Double</a>
(description)	The vertical size of all the content for display in the ScrollViewer.
<b>ExtentWidth</b>	<a href="#">x:Double</a>
(description)	The horizontal size of all the content for display in the ScrollViewer.
<b>HorizontalOffset</b>	<a href="#">x:Double</a>
(description)	A value that contains the horizontal offset of the scrolled content.
<b>HorizontalScrollBarVisibility</b>	<a href="#">ScrollBarVisibility</a>
(description)	A value that indicates whether a horizontal ScrollBar should be displayed.
<b>ScrollableHeight</b>	<a href="#">x:Double</a>
(description)	A value that represents the vertical size of the area that can be scrolled; the difference between the width of the extent and the width of the viewport.
<b>ScrollableWidth</b>	<a href="#">x:Double</a>
(description)	A value that represents the horizontal size of the area that can be scrolled; the difference between the width of the extent and the width of the viewport..
<b>VerticalOffset</b>	<a href="#">x:Double</a>
(description)	A value that contains the vertical offset of the scrolled content.
<b>VerticalScrollBarVisibility</b>	<a href="#">ScrollBarVisibility</a>
(description)	A value that indicates whether a vertical ScrollBar should be displayed.
<b>ViewportHeight</b>	<a href="#">x:Double</a>
(description)	A value that contains the vertical size of the viewable content.
<b>ViewportWidth</b>	<a href="#">x:Double</a>
(description)	A value that contains the horizontal size of the viewable content.
<b>(attachable properties)</b>	
<b>ScrollViewer. HorizontalScrollBarVisibility</b>	<a href="#">ScrollBarVisibility</a>
(description)	A value that indicates whether a horizontal ScrollBar should be displayed.
[target type]	<a href="#">DependencyObject</a>
<b>ScrollViewer. VerticalScrollBarVisibility</b>	<a href="#">ScrollBarVisibility</a>
(description)	A value that indicates whether a vertical ScrollBar should be displayed.
[target type]	<a href="#">DependencyObject</a>

## 1.22.31.1.3.1.4 ToolTip

<b>(usage)</b>	<ToolTip />
<b>(description)</b>	Represents a control that creates a pop-up window that displays information for an element in the UI.
<b>[types assignable to]</b>	<a href="#">ToolTip</a> <a href="#">ContentControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Content</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>HorizontalOffset</b>	<a href="#">x:Double</a>
<b>(description)</b>	The horizontal distance between the target origin and the pop-up alignment point.
<b>IsOpen</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether the ToolTip is visible.
<b>VerticalOffset</b>	<a href="#">x:Double</a>
<b>(description)</b>	The vertical distance between the target origin and the pop-up alignment point.
<b>(events)</b>	
<b>Closed</b>	Occurs when a ToolTip is closed and is no longer visible.
<b>Opened</b>	Occurs when a ToolTip becomes visible.

## 1.22.31.1.3.2 ItemsControl

<b>(usage)</b>	<ItemsControl />
<b>(description)</b>	Represents a control that can be used to present a collection of items.
<b>[types assignable to]</b>	<a href="#">ItemsControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Items</a>
<b>[xml lang property]</b>	<a href="#">Language</a>

<b>(properties)</b>	
<b>DisplayMemberPath</b>	<a href="#">x:String</a>
<b>(description)</b>	The name or path of the property that is displayed for each data item.
<b>Items</b>	<a href="#">ItemCollection</a>
<b>(description)</b>	The collection used to generate the content of the control.
<b>[read only]</b>	True
<b>ItemsPanel</b>	<a href="#">ItemsPanelTemplate</a>
<b>(description)</b>	The template that defines the panel that controls the layout of items.
<b>ItemsSource</b>	<a href="#">IEnumerable</a>
<b>(description)</b>	A collection used to generate the content of the ItemsControl.
<b>ItemTemplate</b>	<a href="#">DataTemplate</a>
<b>(description)</b>	The DataTemplate used to display each item.

## 1.22.31.1.3.2.1 Selector

<b>(usage)</b>	None.
<b>(description)</b>	Represents a control that allows a user to select an item from a collection of items.
<b>[types assignable to]</b>	<a href="#">Selector</a> <a href="#">ItemsControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Items</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>SelectedIndex</b>	<a href="#">x:Int32</a>
<b>(description)</b>	The index of the selected item.
<b>SelectedItem</b>	<a href="#">x:Object</a>
<b>(description)</b>	The selected item.
<b>(events)</b>	
<b>SelectionChanged</b>	Occurs when the currently selected item changes.

## 1.22.31.1.3.2.1.1 ComboBox

<b>(usage)</b>	<ComboBox />
<b>(description)</b>	Represents a selection control that combines a non-editable text box and a popup containing a list box that allows users to select an item from a list.
<b>[types assignable to]</b>	<a href="#">ComboBox</a> <a href="#">Selector</a> <a href="#">ItemsControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Items</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>IsDropDownOpen</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether the popup portion of the combo box is currently open.
<b>IsSelectionBoxHighlighted</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether the SelectionBoxItem is highlighted.
<b>ItemContainerStyle</b>	<a href="#">Style</a>
<b>(description)</b>	The style applied to the container generated for each item in the combo box.
<b>MaxDropDownHeight</b>	<a href="#">x:Double</a>
<b>(description)</b>	The maximum height of the popup that lists combo box items.
<b>SelectionBoxItem</b>	<a href="#">x:Object</a>
<b>(description)</b>	The item displayed in the selection box.
<b>SelectionBoxItemTemplate</b>	<a href="#">DataTemplate</a>
<b>(description)</b>	The template applied to the selection box content.
<b>(events)</b>	
<b>DropDownClosed</b>	Occurs when the popup portion of the combo box closes.
<b>DropDownOpened</b>	Occurs when the popup portion of the combo box opens.

## 1.22.31.1.3.2.1.2 ListBox

<b>(usage)</b>	<ListBox />
<b>(description)</b>	Contains a list of selectable items.
<b>[types assignable to]</b>	<a href="#">ListBox</a> <a href="#">Selector</a> <a href="#">ItemsControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a>

[MS-SLXV] – v0.9

Silverlight Xaml Vocabulary Specification 2008

Copyright © 2008 Microsoft Corporation.

Release: October 2008

	<a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Items</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
ItemContainerStyle	<a href="#">Style</a>
(description)	The style that is used when rendering the item containers.

### 1.22.31.1.3.3 PasswordBox

(usage)	<PasswordBox />
(description)	Represents a control for entering passwords.
[types assignable to]	<a href="#">PasswordBox</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
FontSource	<a href="#">FontSource</a>
(description)	The font source that is applied to the password box for rendering content.
MaxLength	<a href="#">x:Int32</a>
(description)	The maximum length for passwords to be handled by this PasswordBox.
Password	<a href="#">x:String</a>
(description)	The password currently held by the PasswordBox.
PasswordChar	<a href="#">x:Char</a>
(description)	The masking character for the PasswordBox.
SelectionBackground	<a href="#">Brush</a>
(description)	The brush used to render the background for the selected text.
SelectionForeground	<a href="#">Brush</a>
(description)	The brush used for the selected text in the PasswordBox.
(events)	
PasswordChanged	Occurs when the value of the Password property changes.

## 1.22.31.1.3.4 RangeBase

<b>(usage)</b>	None.
<b>(description)</b>	Represents an element that has a value within a specific range, such as the ProgressBar, ScrollBar, and Slider controls.
<b>[types assignable to]</b>	<a href="#">RangeBase</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>LargeChange</b>	<a href="#">x:Double</a>
<b>(description)</b>	A value to be added to or subtracted from the Value of a RangeBase control.
<b>Maximum</b>	<a href="#">x:Double</a>
<b>(description)</b>	The highest possible Value of the range element.
<b>Minimum</b>	<a href="#">x:Double</a>
<b>(description)</b>	The Minimum possible Value of the range element.
<b>SmallChange</b>	<a href="#">x:Double</a>
<b>(description)</b>	A Value to be added to or subtracted from the Value of a RangeBase control.
<b>Value</b>	<a href="#">x:Double</a>
<b>(description)</b>	The current setting of the range control, which may be coerced.
<b>(events)</b>	
<b>ValueChanged</b>	Occurs when the range value changes.

### 1.22.31.1.3.4.1 ProgressBar

<b>(usage)</b>	<ProgressBar />
<b>(description)</b>	Represents a control that indicates the progress of an operation.
<b>[types assignable to]</b>	<a href="#">ProgressBar</a> <a href="#">RangeBase</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>IsIndeterminate</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether the progress bar reports generic progress with a repeating pattern or reports progress based on the Value property.



## 1.22.31.1.3.4.2 ScrollBar

<b>(usage)</b>	<ScrollBar />
<b>(description)</b>	Represents a control that provides a scroll bar that has a sliding Thumb whose position corresponds to a value.
<b>[types assignable to]</b>	<a href="#">ScrollBar</a> <a href="#">RangeBase</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>Orientation</b>	<a href="#">Orientation</a>
<b>(description)</b>	Whether the ScrollBar is displayed horizontally or vertically.
<b>ViewportSize</b>	<a href="#">x:Double</a>
<b>(description)</b>	The amount of the scrollable content that is currently visible.
<b>(events)</b>	
<b>Scroll</b>	Occurs one or more times as content scrolls in a ScrollBar when the user moves the Thumb by using the mouse.

## 1.22.31.1.3.4.3 Slider

<b>(usage)</b>	<Slider />
<b>(description)</b>	Represents a control that lets the user select from a range of values by moving a Thumb control along a track.
<b>[types assignable to]</b>	<a href="#">Slider</a> <a href="#">RangeBase</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>IsDirectionReversed</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates the direction of increasing value.
<b>IsFocused</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value indicating whether the slider control has focus.
<b>Orientation</b>	<a href="#">Orientation</a>

(description)	The orientation of a Slider.
---------------	------------------------------

## 1.22.31.1.3.5 TextBox

(usage)	<TextBox />
---------	-------------

(description)	Represents a control that can be used to display single-format, multi-line text.
---------------	--

[types assignable to]	<a href="#">TextBox</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
-----------------------	--

[xml lang property]	<a href="#">Language</a>
---------------------	--------------------------

(properties)	
--------------	--

<b>AcceptsReturn</b>	<a href="#">x:Boolean</a>
----------------------	---------------------------

(description)	The value that determines whether the text box allows and displays the newline or return characters.
---------------	--

<b>FontSource</b>	<a href="#">FontSource</a>
-------------------	----------------------------

(description)	The font source that is applied to the TextBox for rendering content.
---------------	---

<b>HorizontalScrollBarVisibility</b>	<a href="#">ScrollBarVisibility</a>
--------------------------------------	-------------------------------------

(description)	The visibility of the horizontal scroll bar.
---------------	--

<b>IsReadOnly</b>	<a href="#">x:Boolean</a>
-------------------	---------------------------

(description)	The value that determines if the user can change the text in the text box.
---------------	--

<b>MaxLength</b>	<a href="#">x:Int32</a>
------------------	-------------------------

(description)	The value that determines the maximum number of characters allowed for user input.
---------------	--

<b>SelectedText</b>	<a href="#">x:String</a>
---------------------	--------------------------

(description)	The content of the current selection in the text box.
---------------	---

<b>SelectionBackground</b>	<a href="#">Brush</a>
----------------------------	-----------------------

(description)	The brush that fills the background of the selected text.
---------------	---

<b>SelectionForeground</b>	<a href="#">Brush</a>
----------------------------	-----------------------

(description)	The brush used for the selected text in the text box.
---------------	---

<b>SelectionLength</b>	<a href="#">x:Int32</a>
------------------------	-------------------------

(description)	The number of characters in the current selection in the text box.
---------------	--

<b>SelectionStart</b>	<a href="#">x:Int32</a>
-----------------------	-------------------------

(description)	The starting position of the text selected in the text box.
---------------	---

<b>Text</b>	<a href="#">x:String</a>
<b>(description)</b>	The text contents of the text box.
<b>TextAlignment</b>	<a href="#">TextAlignment</a>
<b>(description)</b>	How the text should be aligned in the text box.
<b>TextWrapping</b>	<a href="#">TextWrapping</a>
<b>(description)</b>	How line breaking occurs if a line of text extends beyond the available width of the text box.
<b>VerticalScrollBarVisibility</b>	<a href="#">ScrollBarVisibility</a>
<b>(description)</b>	The visibility of the vertical scroll bar.
<b>(events)</b>	
<b>SelectionChanged</b>	Occurs when the text selection has changed.
<b>TextChanged</b>	Occurs when content changes in the text box.

## 1.22.31.1.3.6 Thumb

<b>(usage)</b>	<Thumb />
<b>(description)</b>	Represents a control that can be dragged by the user.
<b>[types assignable to]</b>	<a href="#">Thumb Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>IsDragging</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	Whether the Thumb control has focus and mouse capture.
<b>IsFocused</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	Whether the thumb has focus.
<b>(events)</b>	
<b>DragCompleted</b>	Occurs when the Thumb control loses mouse capture.
<b>DragDelta</b>	Occurs one or more times as the mouse pointer is moved when a Thumb control has logical focus and mouse capture.
<b>DragStarted</b>	Occurs when a Thumb control receives logical focus and mouse capture.

## 1.22.31.1.3.7 UserControl

<b>(usage)</b>	<UserControl />
<b>(description)</b>	Provides the base type for defining a new control that encapsulates related existing controls and provides its own logic.
<b>[types assignable to]</b>	<a href="#">UserControl</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>[is name scope]</b>	True

## 1.22.31.1.4 Glyphs

<b>(usage)</b>	<Glyphs />
<b>(description)</b>	Provides a visual representation of letters, characters, or symbols, in a specific font and style.
<b>[types assignable to]</b>	<a href="#">Glyphs</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>Fill</b>	<a href="#">Brush</a>
<b>(description)</b>	The Brush that is used to render the glyphs.
<b>FontRenderingEmSize</b>	<a href="#">x:Double</a>
<b>(description)</b>	The em size used for rendering the glyphs.
<b>FontUri</b>	<a href="#">x:Uri</a>
<b>(description)</b>	The location of the font used for rendering the glyphs.
<b>Indices</b>	<a href="#">x:String</a>
<b>(description)</b>	The glyph indices for the glyphs.
<b>OriginX</b>	<a href="#">x:Double</a>
<b>(description)</b>	The x origin for the glyphs.
<b>OriginY</b>	<a href="#">x:Double</a>
<b>(description)</b>	The y origin for the glyphs.
<b>StyleSimulations</b>	<a href="#">StyleSimulations</a>
<b>(description)</b>	The style simulations applied to the glyphs.

<b>UnicodeString</b>	<a href="#">x:String</a>
(description)	The Unicode string to render in glyphs.

## 1.22.31.1.5 Image

(usage)	<Image />
(description)	Represents a control that displays an image.
[types assignable to]	<a href="#">Image</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
Source	<a href="#">ImageSource</a>
(description)	The source for the image.
Stretch	<a href="#">Stretch</a>
(description)	A value that describes how an Image should be stretched to fill the destination rectangle.
(events)	
ImageFailed	Occurs when there is an error associated with image retrieval or format.

## 1.22.31.1.6 ItemsPresenter

(usage)	<ItemsPresenter />
(description)	Specifies where items are placed in a control, usually an ItemsControl.
[types assignable to]	<a href="#">ItemsPresenter</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>

## 1.22.31.1.7 MediaElement

(usage)	<MediaElement />
(description)	Represents an object that contains audio, video, or both.
[types assignable to]	<a href="#">MediaElement</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>

<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>Attributes</b>	<a href="#">DictionaryOfStringStringOfString</a>
<b>(description)</b>	The collection of attributes that corresponds to the current entry in the ASX file that Source is set to.
<b>[read only]</b>	True
<b>AudioStreamIndex</b>	<a href="#">NullableOfInt32</a>
<b>(description)</b>	The index of the audio stream that plays along with the video component. The collection of audio streams is composed at run time and represents all audio streams available within the media file.
<b>AutoPlay</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether media will begin playback automatically when the Source property is set.
<b>Balance</b>	<a href="#">x:Double</a>
<b>(description)</b>	A ratio of volume across stereo speakers.
<b>BufferingTime</b>	<a href="#">x:TimeSpan</a>
<b>(description)</b>	The amount of time to buffer.
<b>IsMuted</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value indicating whether the audio is muted.
<b>LicenseAcquirer</b>	<a href="#">LicenseAcquirer</a>
<b>(description)</b>	The LicenseAcquirer associated with the MediaElement. The LicenseAcquirer handles acquiring licenses for DRM encrypted content.
<b>Markers</b>	<a href="#">TimelineMarkerCollection</a>
<b>(description)</b>	The collection of timeline markers associated with the currently loaded media file.
<b>[read only]</b>	True
<b>Position</b>	<a href="#">x:TimeSpan</a>
<b>(description)</b>	The current position of progress through the media's playback time.
<b>Source</b>	<a href="#">x:Uri</a>
<b>(description)</b>	A media source on the MediaElement.
<b>Stretch</b>	<a href="#">Stretch</a>
<b>(description)</b>	A Stretch value that describes how a MediaElement fills the destination rectangle.

<b>Volume</b>	<a href="#">x:Double</a>
<b>(description)</b>	The media's volume.
<b>(events)</b>	
<b>BufferingProgressChanged</b>	Occurs when the BufferingProgress property changes.
<b>CurrentStateChanged</b>	Occurs when the value of the CurrentState property changes.
<b>DownloadProgressChanged</b>	Occurs when the DownloadProgress property has changed.
<b>MarkerReached</b>	Occurs when a timeline marker is encountered during media playback.
<b>MediaEnded</b>	Occurs when the MediaElement is no longer playing audio or video.
<b>MediaFailed</b>	Occurs when there is an error associated with the media Source.
<b>MediaOpened</b>	Occurs when the media stream has been validated and opened, and the file headers have been read.

## 1.22.31.1.8 MultiScaleImage

<b>(usage)</b>	<code>&lt;MultiScaleImage /&gt;</code>
<b>(description)</b>	This object enables users to open a multi-resolution image which can be zoomed in on and panned across. This is one of the key objects of Silverlight's Deep Zoom technology.
<b>[types assignable to]</b>	<a href="#">MultiScaleImage</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>Source</b>	<a href="#">MultiScaleTileSource</a>
<b>(description)</b>	The URI of the image to be used by the MultiScaleImage.
<b>UseSprings</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	Whether the MultiScaleImage uses spring animations.
<b>ViewportOrigin</b>	<a href="#">Point</a>
<b>(description)</b>	The top-left corner of the area of the image to be displayed.
<b>ViewportWidth</b>	<a href="#">x:Double</a>
<b>(description)</b>	The width of the area of the image displayed.
<b>(events)</b>	
<b>ImageFailed</b>	Occurs if the download of a tile times out or fails for another reason.
<b>ImageOpenFailed</b>	Occurs if the first piece of metadata used to open the image fails. If this event occurs no parts of the image will open successfully.

<b>ImageOpenSucceeded</b>	Occurs when the first piece of metadata that is needed to load the rest of the tiles opens.
<b>MotionFinished</b>	Occurs when the zoom or pan animation ends.
<b>ViewportChanged</b>	Occurs when the viewport (the area of the image displayed) changes.

## 1.22.31.1.9 Panel

<b>(usage)</b>	None.
<b>(description)</b>	Provides a base type for all Panel elements. Use Panel elements to position and arrange child objects in Silverlight-based applications.
<b>[types assignable to]</b>	<a href="#">Panel</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Children</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(properties)</b>	
<b>Background</b>	<a href="#">Brush</a>
<b>(description)</b>	A Brush that is used to fill the panel.
<b>Children</b>	<a href="#">UIElementCollection</a>
<b>(description)</b>	The collection of child elements of the panel.
<b>[read only]</b>	True

### 1.22.31.1.9.1 Canvas

<b>(usage)</b>	<Canvas />
<b>(description)</b>	Defines an area within which you can explicitly position child objects by using coordinates that are relative to the area.
<b>[types assignable to]</b>	<a href="#">Canvas</a> <a href="#">Panel</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Children</a>
<b>[xml lang property]</b>	<a href="#">Language</a>
<b>(attachable properties)</b>	
<b>Canvas.Left</b>	<a href="#">x:Double</a>



(description)	The distance between the left side of an object and the left side of its parent Canvas.
[target type]	<a href="#">UIElement</a>
Canvas.Top	<a href="#">x:Double</a>
(description)	The distance between the top of an element and the top of its parent Canvas.
[target type]	<a href="#">UIElement</a>
Canvas.ZIndex	<a href="#">x:Int32</a>
(description)	The z-order rendering behavior of objects in a Canvas.
[target type]	<a href="#">UIElement</a>

## 1.22.31.1.9.1.1 InkPresenter

(usage)	<code>&lt;InkPresenter /&gt;</code>
(description)	Implements a rectangular surface that displays ink strokes.
[types assignable to]	<a href="#">InkPresenter</a> <a href="#">Canvas</a> <a href="#">Panel</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Children</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
Strokes	<a href="#">StrokeCollection</a>
(description)	The strokes that the InkPresenter displays.

## 1.22.31.1.9.2 Grid

(usage)	<code>&lt;Grid /&gt;</code>
(description)	Defines a flexible grid area that consists of columns and rows.
[types assignable to]	<a href="#">Grid</a> <a href="#">Panel</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Children</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
ColumnDefinitions	<a href="#">ColumnDefinitionCollection</a>

<b>(description)</b>	A ColumnDefinitionCollection defined on this instance of Grid.
<b>[read only]</b>	True
<b>RowDefinitions</b>	<a href="#">RowDefinitionCollection</a>
<b>(description)</b>	A RowDefinitionCollection defined on this instance of Grid.
<b>[read only]</b>	True
<b>ShowGridLines</b>	<a href="#">x:Boolean</a>
<b>(description)</b>	A value that indicates whether grid lines are visible within this Grid.
<b>(attachable properties)</b>	
<b>Grid.Column</b>	<a href="#">x:Int32</a>
<b>(description)</b>	A value that indicates which column child content within a Grid should appear in.
<b>[target type]</b>	<a href="#">FrameworkElement</a>
<b>Grid.ColumnSpan</b>	<a href="#">x:Int32</a>
<b>(description)</b>	A value that indicates the total number of columns that child content spans within a Grid.
<b>[target type]</b>	<a href="#">FrameworkElement</a>
<b>Grid.Row</b>	<a href="#">x:Int32</a>
<b>(description)</b>	A value that indicates which row child content within a Grid should appear in.
<b>[target type]</b>	<a href="#">FrameworkElement</a>
<b>Grid.RowSpan</b>	<a href="#">x:Int32</a>
<b>(description)</b>	A value that indicates the total number of rows that child content spans within a Grid.
<b>[target type]</b>	<a href="#">FrameworkElement</a>

### 1.22.31.1.9.3 StackPanel

<b>(usage)</b>	<StackPanel />
<b>(description)</b>	Arranges child elements into a single line that can be oriented horizontally or vertically.
<b>[types assignable to]</b>	<a href="#">StackPanel</a> <a href="#">Panel</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
<b>[content property]</b>	<a href="#">Children</a>

[xml lang property]	<a href="#">Language</a>
(properties)	
Orientation	<a href="#">Orientation</a>
(description)	The dimension by which child elements are stacked.

## 1.22.31.1.10 Popup

(usage)	<Popup />
(description)	Displays content on top of existing Silverlight content, within the bounds of the Silverlight control.
[types assignable to]	<a href="#">Popup</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Child</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
Child	<a href="#">UIElement</a>
(description)	The content to be hosted in the popup.
HorizontalOffset	<a href="#">x:Double</a>
(description)	The distance between the left side of the Silverlight control and the left side of the popup.
IsOpen	<a href="#">x:Boolean</a>
(description)	Whether the popup is currently displaying on the screen.
VerticalOffset	<a href="#">x:Double</a>
(description)	The distance between the top of the Silverlight control and the top of the popup.
(events)	
Closed	Occurs when the IsOpen property is set to false.
Opened	Occurs when the IsOpen property is set to true.

## 1.22.31.1.11 Shape

(usage)	None.
(description)	Provides a base type for shape elements, such as Ellipse, Polygon, and

	Rectangle.
[types assignable to]	<a href="#">Shape</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>
<b>(properties)</b>	
<b>Fill</b>	<a href="#">Brush</a>
<b>(description)</b>	The Brush that specifies how to paint the interior of the shape.
<b>Stretch</b>	<a href="#">Stretch</a>
<b>(description)</b>	A Stretch enumeration value that describes how the shape fills its allocated space.
<b>Stroke</b>	<a href="#">Brush</a>
<b>(description)</b>	The Brush that specifies how the Shape outline is painted.
<b>StrokeDashArray</b>	<a href="#">DoubleCollection</a>
<b>(description)</b>	A collection of Double values that indicate the pattern of dashes and gaps that is used to outline shapes.
<b>StrokeDashCap</b>	<a href="#">PenLineCap</a>
<b>(description)</b>	A PenLineCap enumeration value that specifies how the ends of a dash are drawn.
<b>StrokeDashOffset</b>	<a href="#">x:Double</a>
<b>(description)</b>	A Double that specifies the distance within the dash pattern where a dash begins.
<b>StrokeEndLineCap</b>	<a href="#">PenLineCap</a>
<b>(description)</b>	A PenLineCap enumeration value that describes the Shape at the end of a line.
<b>StrokeLineJoin</b>	<a href="#">PenLineJoin</a>
<b>(description)</b>	A PenLineJoin enumeration value that specifies the type of join that is used at the vertices of a Shape.
<b>StrokeMiterLimit</b>	<a href="#">x:Double</a>
<b>(description)</b>	A limit on the ratio of the miter length to half the StrokeThickness of a Shape element.
<b>StrokeStartLineCap</b>	<a href="#">PenLineCap</a>
<b>(description)</b>	A PenLineCap enumeration value that describes the Shape at the start of a Stroke.
<b>StrokeThickness</b>	<a href="#">x:Double</a>
<b>(description)</b>	The width of the Shape stroke outline.

## 1.22.31.1.11.1 Ellipse

(usage)	<Ellipse />
(description)	Draws an ellipse.
[types assignable to]	<a href="#">Ellipse</a> <a href="#">Shape</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>

## 1.22.31.1.11.2 Line

(usage)	<Line />
(description)	Draws a straight line between two points.
[types assignable to]	<a href="#">Line</a> <a href="#">Shape</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>

### (properties)

X1 [x:Double](#)

(description) The x-coordinate of the Line start point.

X2 [x:Double](#)

(description) The x-coordinate of the Line end point.

Y1 [x:Double](#)

(description) The y-coordinate of the Line start point.

Y2 [x:Double](#)

(description) The y-coordinate of the Line end point.

## 1.22.31.1.11.3 Path

(usage)	<Path />
(description)	Draws a series of connected lines and curves. The line and curve dimensions are declared through the Data property, and can be specified either with a Path-specific mini-language, or with an object model.
[types assignable to]	<a href="#">Path</a> <a href="#">Shape</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>

[xml lang property]	<a href="#">Language</a>
<b>(properties)</b>	
Data	<a href="#">Geometry</a>
(description)	A Geometry that specifies the shape to be drawn.

## 1.22.31.1.11.4 Polygon

(usage)	<Polygon />
(description)	Draws a polygon, which is a connected series of lines that form a closed shape.
[types assignable to]	<a href="#">Polygon</a> <a href="#">Shape</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>
<b>(properties)</b>	
FillRule	<a href="#">FillRule</a>
(description)	A value that specifies how the interior fill of the shape is determined.
Points	<a href="#">PointCollection</a>
(description)	A collection that contains the vertex points of the polygon.

## 1.22.31.1.11.5 Polyline

(usage)	<Polyline />
(description)	Draws a series of connected straight lines.
[types assignable to]	<a href="#">Polyline</a> <a href="#">Shape</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>
<b>(properties)</b>	
FillRule	<a href="#">FillRule</a>
(description)	A value that specifies how the interior fill of the shape is determined.
Points	<a href="#">PointCollection</a>
(description)	A collection that contains the vertex points of the Polyline.

## 1.22.31.1.11.6 Rectangle

(usage)	<Rectangle />
(description)	Draws a rectangle shape, which can have a stroke and a fill.
[types assignable to]	<a href="#">Rectangle</a> <a href="#">Shape</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
RadiusX	<a href="#">x:Double</a>
(description)	The x-axis radius of the ellipse that is used to round the corners of the rectangle.
RadiusY	<a href="#">x:Double</a>
(description)	The y-axis radius of the ellipse that is used to round the corners of the rectangle.

## 1.22.31.1.12 TextBlock

(usage)	<TextBlock />
(description)	Provides a lightweight control for displaying small amounts of flow content.
[types assignable to]	<a href="#">TextBlock</a> <a href="#">FrameworkElement</a> <a href="#">UIElement</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">Inlines</a>
[xml lang property]	<a href="#">Language</a>
(properties)	
FontFamily	<a href="#">FontFamily</a>
(description)	The preferred top-level font family for the content in this element.
FontSize	<a href="#">x:Double</a>
(description)	The font size for the content in this element.
FontSource	<a href="#">FontSource</a>
(description)	The font source that is applied to the text for rendering content.
FontStretch	<a href="#">FontStretch</a>
(description)	The font stretch for the content in this element.
FontStyle	<a href="#">FontStyle</a>

(description)	The font style for the content in this element.
<b>FontWeight</b>	<a href="#">FontWeight</a>
(description)	The top-level font weight for the TextBlock.
<b>Foreground</b>	<a href="#">Brush</a>
(description)	The Brush to apply to the text contents of the TextBlock.
<b>Inlines</b>	<a href="#">InlineCollection</a>
(description)	The collection of inline text elements within a TextBlock.
[read only]	True
<b>LineHeight</b>	<a href="#">x:Double</a>
(description)	The height of each line of content.
<b>LineStackingStrategy</b>	<a href="#">LineStackingStrategy</a>
(description)	A value that indicates how a line box is determined for each line of text in the TextBlock.
<b>Padding</b>	<a href="#">Thickness</a>
(description)	A value that indicates the thickness of padding space between the boundaries of the content area and the content displayed by a TextBlock.
<b>Text</b>	<a href="#">x:String</a>
(description)	The text contents of a TextBlock.
<b>TextAlignment</b>	<a href="#">TextAlignment</a>
(description)	A value that indicates the horizontal alignment of text content.
<b>TextDecorations</b>	<a href="#">TextDecorationCollection</a>
(description)	A value that specifies the text decorations that are applied to the content in a TextBlock element.
<b>TextWrapping</b>	<a href="#">TextWrapping</a>
(description)	How the TextBlock wraps text.

## 1.22.32 VisualState

(usage)	<VisualState />
(description)	Represents the visual appearance of the control when it is in a specific state.
[types assignable to]	<a href="#">VisualState</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>



[content property]	<a href="#">Storyboard</a>
(properties)	
Storyboard	<a href="#">Storyboard</a>
(description)	A Storyboard that defines the appearance of the control when it is the state that is represented by the VisualState.

### 1.22.33 VisualStateGroup

(usage)	<VisualStateGroup />
(description)	Contains mutually exclusive VisualState objects and VisualTransition objects that are used to go from one state to another.
[types assignable to]	<a href="#">VisualStateGroup</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
[content property]	<a href="#">States</a>
(properties)	
States	<a href="#">IList</a>
(description)	The collection of mutually exclusive VisualState objects.
[read only]	True
Transitions	<a href="#">IList</a>
(description)	The collection of VisualTransition objects.
[read only]	True
(events)	
CurrentStateChanged	Occurs after a control transitions into a different state.
CurrentStateChanging	Occurs when a control begins transitioning into a different state.

### 1.22.34 VisualStateManager

(usage)	<VisualStateManager />
(description)	Manages states and the logic for transitioning between states for controls.
[types assignable to]	<a href="#">VisualStateManager</a> <a href="#">DependencyObject</a> <a href="#">x:Object</a>
(used by)	<a href="#">VisualStateManager</a>
(attachable properties)	

<b>VisualStateManager. CustomVisualStateManager</b>	<a href="#">VisualStateManager</a>
<b>(description)</b>	The VisualStateManager that transitions between the states of a control.
<b>[target type]</b>	<a href="#">FrameworkElement</a>

## 1.23 DependencyProperty

<b>(usage)</b>	None.
<b>(description)</b>	Represents a property that is registered with the Silverlight property system. Dependency properties provide support for value expressions, data binding, animation, and property change notification.
<b>[types assignable to]</b>	<a href="#">DependencyProperty</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Setter</a>

## 1.25 x:Double

<b>link to externally defined type</b>	<a href="#">x:Double</a> , from <a href="#">[MS-XAML]</a>
--	---

## 1.26 DoubleCollection

<b>(usage)</b>	<DoubleCollection> string </DoubleCollection>
<b>(description)</b>	Represents an ordered collection of Double values.
<b>[types assignable to]</b>	<a href="#">DoubleCollection</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Shape</a>
<b>[text syntax]</b>	<a href="#">DoubleCollectionSyntax</a>
<b>[is list]</b>	True
<b>[allowed types]</b>	<a href="#">x:Double</a>

## 1.27 DoubleKeyFrameCollection

<b>(usage)</b>	<DoubleKeyFrameCollection> <a href="#">DoubleKeyFrame</a> * </DoubleKeyFrameCollection>
<b>(description)</b>	Represents a collection of DoubleKeyFrame objects that can be individually accessed by index.
<b>[types assignable to]</b>	<a href="#">DoubleKeyFrameCollection</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">DoubleAnimationUsingKeyFrames</a>
<b>[is list]</b>	True
<b>[allowed types]</b>	<a href="#">DoubleKeyFrame</a>

## 1.28 Duration

<b>(usage)</b>	<Duration> string </Duration>
<b>(description)</b>	Represents the duration of time that a Timeline is active.
<b>[types assignable to]</b>	<a href="#">Duration</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Duration</a> <a href="#">Timeline</a>
<b>[is default constructible]</b>	False
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">DurationSyntax</a>

## 1.29 FillBehavior

<b>(usage)</b>	<a href="#">HoldEnd</a>   <a href="#">Stop</a>
<b>(description)</b>	Specifies how a Timeline behaves when it is outside its active period but its parent is inside its active or hold period.
<b>[types assignable to]</b>	<a href="#">FillBehavior</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Timeline</a>
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">FillBehaviorSyntax</a>

## 1.30 FillRule

(usage)	<a href="#">EvenOdd</a>   <a href="#">Nonzero</a>
(description)	Specifies how the intersecting areas of PathFigure objects contained in a Geometry are combined to form the area of the Geometry.
[types assignable to]	<a href="#">FillRule</a> <a href="#">x:Object</a>
(used by)	<a href="#">GeometryGroup</a> <a href="#">PathGeometry</a> <a href="#">Polygon</a> <a href="#">Polyline</a>
[is nullable]	False
[text syntax]	<a href="#">FillRuleSyntax</a>

## 1.31 FontFamily

(usage)	<FontFamily> string </FontFamily>
(description)	Represents a family of related fonts.
[types assignable to]	<a href="#">FontFamily</a> <a href="#">x:Object</a>
(used by)	<a href="#">Control</a> <a href="#">Inline</a> <a href="#">TextBlock</a>
[text syntax]	<a href="#">FontFamilySyntax</a>

## 1.32 FontSource

(usage)	None.
(description)	Represents one or more fonts created from a stream.
[types assignable to]	<a href="#">FontSource</a> <a href="#">x:Object</a>
(used by)	<a href="#">PasswordBox</a> <a href="#">TextBlock</a> <a href="#">TextBox</a>

## 1.33 FontStretch

(usage)	<a href="#">Condensed</a>   <a href="#">Expanded</a>   <a href="#">ExtraCondensed</a>   <a href="#">ExtraExpanded</a>   <a href="#">Normal</a>   <a href="#">SemiCondensed</a>   <a href="#">SemiExpanded</a>   <a href="#">UltraCondensed</a>   <a href="#">UltraExpanded</a>   ...
---------	--

(description)	Describes the degree to which a font has been stretched, compared to the normal aspect ratio of that font.
[types assignable to]	<a href="#">FontStretch x:Object</a>
(used by)	<a href="#">Control</a> <a href="#">FontStretches</a> <a href="#">Inline</a> <a href="#">TextBlock</a>
[is default constructible]	False
[is nullable]	False
[text syntax]	<a href="#">FontStretchSyntax</a>

## 1.34 FontStretches

(usage)	{x:Static <a href="#">FontStretches.StaticPropertyName</a> }
(description)	Provides a set of predefined font stretches as static property values.
[types assignable to]	<a href="#">FontStretches x:Object</a>

## 1.35 FontStyle

(usage)	<a href="#">Normal</a>   <a href="#">Italic</a>
(description)	Represents the style of a font face (for instance, as normal or italic).
[types assignable to]	<a href="#">FontStyle x:Object</a>
(used by)	<a href="#">Control</a> <a href="#">FontStyles</a> <a href="#">Inline</a> <a href="#">TextBlock</a>
[is default constructible]	False
[is nullable]	False
[text syntax]	<a href="#">FontStyleSyntax</a>

## 1.36 FontStyles

(usage)	{x:Static <a href="#">FontStyles.StaticPropertyName</a> }
(description)	Provides a set of predefined font styles as static property values.
[types assignable to]	<a href="#">FontStyles x:Object</a>

## 1.37 FontWeight

(usage)	<a href="#">Black</a>   <a href="#">Bold</a>   <a href="#">ExtraBlack</a>   <a href="#">ExtraBold</a>   <a href="#">ExtraLight</a>   <a href="#">Light</a>   <a href="#">Medium</a>   <a href="#">Normal</a>   <a href="#">SemiBold</a>   ...
(description)	Refers to the density of a typeface, in terms of the lightness or heaviness of the strokes.
[types assignable to]	<a href="#">FontWeight</a> <a href="#">x:Object</a>
(used by)	<a href="#">Control</a> <a href="#">FontWeights</a> <a href="#">Inline</a> <a href="#">TextBlock</a>
[is default constructible]	False
[is nullable]	False
[text syntax]	<a href="#">FontWeightSyntax</a>

## 1.38 FontWeights

(usage)	{x:Static <a href="#">FontWeights.StaticPropertyName</a> }
(description)	Provides a set of predefined font weights as static property values.
[types assignable to]	<a href="#">FontWeights</a> <a href="#">x:Object</a>

## 1.39 GeometryCollection

(usage)	< <a href="#">GeometryCollection</a> > <a href="#">Geometry</a> * </ <a href="#">GeometryCollection</a> >
(description)	Represents a collection of <a href="#">Geometry</a> objects.
[types assignable to]	<a href="#">GeometryCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">GeometryGroup</a>
[is list]	True
[allowed types]	<a href="#">Geometry</a>

## 1.40 GradientSpreadMethod

<b>(usage)</b>	<a href="#">Pad</a>   <a href="#">Reflect</a>   <a href="#">Repeat</a>
<b>(description)</b>	Specifies how to draw the gradient outside a gradient brush's gradient vector or space.
<b>[types assignable to]</b>	<a href="#">GradientSpreadMethod x:Object</a>
<b>(used by)</b>	<a href="#">GradientBrush</a>
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">GradientSpreadMethodSyntax</a>

## 1.41 GradientStopCollection

<b>(usage)</b>	<GradientStopCollection> <a href="#">GradientStop</a> * </GradientStopCollection>
<b>(description)</b>	Represents a collection of GradientStop objects that can be individually accessed by index.
<b>[types assignable to]</b>	<a href="#">GradientStopCollection x:Object</a>
<b>(used by)</b>	<a href="#">GradientBrush</a>
<b>[is list]</b>	True
<b>[allowed types]</b>	<a href="#">GradientStop</a>

## 1.42 GridLength

<b>(usage)</b>	<GridLength> string </GridLength>
<b>(description)</b>	Represents the length of elements that explicitly support Star unit types.
<b>[types assignable to]</b>	<a href="#">GridLength x:Object</a>
<b>(used by)</b>	<a href="#">ColumnDefinition</a> <a href="#">GridLength</a> <a href="#">RowDefinition</a>
<b>[is default constructible]</b>	False
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">GridLengthSyntax</a>

## 1.43 HorizontalAlignment

(usage)	<a href="#">Left</a>   <a href="#">Center</a>   <a href="#">Right</a>   <a href="#">Stretch</a>
(description)	Indicates where an element should be displayed on the horizontal axis relative to the allocated layout slot of the parent element.
[types assignable to]	<a href="#">HorizontalAlignment</a> <a href="#">x:Object</a>
(used by)	<a href="#">Control</a> <a href="#">FrameworkElement</a>
[is nullable]	False
[text syntax]	<a href="#">HorizontalAlignmentSyntax</a>

## 1.44 InlineCollection

(usage)	None.
(description)	Represents a collection of Inline elements.
[types assignable to]	<a href="#">InlineCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">TextBlock</a>
[whitespace significant collection]	True
[is list]	True
[allowed types]	<a href="#">x:String</a> <a href="#">Inline</a>

## 1.45 x:Int32

link to externally defined type	<a href="#">x:Int32</a> , from <a href="#">[MS-XAML]</a>
---------------------------------	--

## 1.46 ItemCollection

(usage)	None.
(description)	Holds the list of items that constitute the content of an ItemsControl.
[types assignable to]	<a href="#">ItemCollection</a> <a href="#">x:Object</a>



(used by)	<a href="#">ItemsControl</a>
[is list]	True
[allowed types]	<a href="#">x:Object</a>

## 1.47 Keyboard

(usage)	{x:Static Keyboard. <i>StaticPropertyName</i> }
(description)	Represents the keyboard device.
[types assignable to]	<a href="#">Keyboard</a> <a href="#">x:Object</a>

## 1.48 KeyboardNavigationMode

(usage)	<a href="#">Local</a>   <a href="#">Cycle</a>   <a href="#">Once</a>
(description)	Specifies the tabbing behavior across tab stops for a Silverlight tabbing sequence within a container.
[types assignable to]	<a href="#">KeyboardNavigationMode</a> <a href="#">x:Object</a>
(used by)	<a href="#">Control</a>
[is nullable]	False
[text syntax]	<a href="#">KeyboardNavigationModeSyntax</a>

## 1.49 KeyTime

(usage)	<KeyTime> string </KeyTime>
(description)	Specifies when a particular key frame should take place during an animation.
[types assignable to]	<a href="#">KeyTime</a> <a href="#">x:Object</a>
(used by)	<a href="#">ColorKeyFrame</a> <a href="#">DoubleKeyFrame</a> <a href="#">KeyTime</a> <a href="#">ObjectKeyFrame</a> <a href="#">PointKeyFrame</a>
[is default constructible]	False
[is nullable]	False
[text syntax]	<a href="#">KeyTimeSyntax</a>

## 1.50 LicenseAcquirer

(usage)	<LicenseAcquirer />
(description)	This type handles acquiring licenses for DRM encrypted content.
[types assignable to]	<a href="#">LicenseAcquirer x:Object</a>
(used by)	<a href="#">MediaElement</a>
(properties)	
LicenseServerUriOverride	<a href="#">x:Uri</a>
(description)	A URI value that overrides whatever the license server URI is in the content header.

## 1.51 LineStackingStrategy

(usage)	<a href="#">MaxHeight</a>   <a href="#">BlockLineHeight</a>
(description)	Describes the mechanism by which a line box is determined for each line.
[types assignable to]	<a href="#">LineStackingStrategy x:Object</a>
(used by)	<a href="#">TextBlock</a>
[is nullable]	False
[text syntax]	<a href="#">LineStackingStrategySyntax</a>

## 1.52 Matrix

(usage)	<Matrix> string </Matrix>
(description)	Represents a 3x3 affine transformation matrix used for transformations in two-dimensional space.
[types assignable to]	<a href="#">Matrix x:Object</a>
(used by)	<a href="#">Matrix</a> <a href="#">MatrixTransform</a>
[is default constructible]	False
[is nullable]	False

[text syntax]	<a href="#">MatrixSyntax</a>
(properties)	
M11	<a href="#">x:Double</a>
(description)	The value of the first row and first column of this Matrix structure.
M12	<a href="#">x:Double</a>
(description)	The value of the first row and second column of this Matrix structure.
M21	<a href="#">x:Double</a>
(description)	The value of the second row and first column of this Matrix structure.
M22	<a href="#">x:Double</a>
(description)	The value of the second row and second column of this Matrix structure.
OffsetX	<a href="#">x:Double</a>
(description)	The value of the third row and first column of this Matrix structure.
OffsetY	<a href="#">x:Double</a>
(description)	The value of the third row and second column of this Matrix structure.

## 1.53.1 x:XamlType

link to externally defined type [x:XamlType](#), from [\[MS-XAML\]](#)

## 1.54 ModifierKeys

(usage)	<a href="#">None</a>   [ <a href="#">Alt</a>   <a href="#">Control</a>   <a href="#">Shift</a>   <a href="#">Windows</a>   <a href="#">Apple</a> ]*
(description)	Specifies the set of modifier keys.
[types assignable to]	<a href="#">ModifierKeys</a> <a href="#">x:Object</a>
(used by)	<a href="#">Keyboard</a>
[is nullable]	False
[text syntax]	<a href="#">ModifierKeysSyntax</a>

## 1.55 ObjectKeyFrameCollection

<b>(usage)</b>	<ObjectKeyFrameCollection> <a href="#">ObjectKeyFrame</a> * </ObjectKeyFrameCollection>
<b>(description)</b>	Represents a collection of ObjectKeyFrame objects that can be individually accessed by index.
<b>[types assignable to]</b>	<a href="#">ObjectKeyFrameCollection</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">ObjectAnimationUsingKeyFrames</a>
<b>[is list]</b>	True
<b>[allowed types]</b>	<a href="#">ObjectKeyFrame</a>

## 1.57 Orientation

<b>(usage)</b>	<a href="#">Vertical</a>   <a href="#">Horizontal</a>
<b>(description)</b>	Defines the different orientations that a control or layout can have.
<b>[types assignable to]</b>	<a href="#">Orientation</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">ScrollBar</a> <a href="#">Slider</a> <a href="#">StackPanel</a>
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">OrientationSyntax</a>

## 1.58 PathFigureCollection

<b>(usage)</b>	<PathFigureCollection> <a href="#">PathFigure</a> * </PathFigureCollection>
<b>(description)</b>	Represents a collection of PathFigure objects that collectively make up the geometry of a PathGeometry.
<b>[types assignable to]</b>	<a href="#">PathFigureCollection</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">PathGeometry</a>
<b>[is list]</b>	True
<b>[allowed types]</b>	<a href="#">PathFigure</a>

## 1.59 PathSegmentCollection

<b>(usage)</b>	<PathSegmentCollection> <a href="#">PathSegment</a> * </PathSegmentCollection>
<b>(description)</b>	Represents a collection of PathSegment objects that can be individually accessed by index.
<b>[types assignable to]</b>	<a href="#">PathSegmentCollection</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">PathFigure</a>
<b>[is list]</b>	True
<b>[allowed types]</b>	<a href="#">PathSegment</a>

## 1.60 PenLineCap

<b>(usage)</b>	<a href="#">Flat</a>   <a href="#">Square</a>   <a href="#">Round</a>   <a href="#">Triangle</a>
<b>(description)</b>	Describes the shape at the end of a line or segment.
<b>[types assignable to]</b>	<a href="#">PenLineCap</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Shape</a>
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">PenLineCapSyntax</a>

## 1.61 PenLineJoin

<b>(usage)</b>	<a href="#">Miter</a>   <a href="#">Bevel</a>   <a href="#">Round</a>
<b>(description)</b>	Describes the shape that joins two lines or segments.
<b>[types assignable to]</b>	<a href="#">PenLineJoin</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">Shape</a>
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">PenLineJoinSyntax</a>

## 1.62 Point

<b>(usage)</b>	<Point> string </Point>
<b>(description)</b>	Represents an x- and y-coordinate pair in two-dimensional space. Can also

represent a logical point for certain property usages.

<b>[types assignable to]</b>	<a href="#">Point</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">ArcSegment</a> <a href="#">BezierSegment</a> <a href="#">EllipseGeometry</a> <a href="#">KeySpline</a> <a href="#">LinearGradientBrush</a> <a href="#">LineGeometry</a> <a href="#">LineSegment</a> <a href="#">MultiScaleImage</a> <a href="#">PathFigure</a> <a href="#">PointCollection</a> <a href="#">PointKeyFrame</a> <a href="#">QuadraticBezierSegment</a> <a href="#">RadialGradientBrush</a> <a href="#">UIElement</a>
<b>[is default constructible]</b>	False
<b>[is nullable]</b>	False
<b>[text syntax]</b>	<a href="#">PointSyntax</a>
<b>(properties)</b>	
<b>X</b>	<a href="#">x:Double</a>
<b>(description)</b>	The X-coordinate value of this Point structure.
<b>Y</b>	<a href="#">x:Double</a>
<b>(description)</b>	The Y-coordinate value of this Point.

## 1.63 PointCollection

<b>(usage)</b>	<PointCollection> string </PointCollection>
<b>(description)</b>	Represents a collection of Point values that can be individually accessed by index.
<b>[types assignable to]</b>	<a href="#">PointCollection</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">PolyBezierSegment</a> <a href="#">Polygon</a> <a href="#">Polyline</a> <a href="#">PolyLineSegment</a> <a href="#">PolyQuadraticBezierSegment</a>
<b>[text syntax]</b>	<a href="#">PointCollectionSyntax</a>
<b>[is list]</b>	True
<b>[allowed types]</b>	<a href="#">Point</a>

## 1.64 PointKeyFrameCollection

<b>(usage)</b>	<PointKeyFrameCollection> <a href="#">PointKeyFrame</a> * </PointKeyFrameCollection>
<b>(description)</b>	Represents a collection of PointKeyFrame objects that can be individually accessed by index.

[types assignable to]	<a href="#">PointKeyFrameCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">PointAnimationUsingKeyFrames</a>
[is list]	True
[allowed types]	<a href="#">PointKeyFrame</a>

## 1.65 PropertyPath

(usage)	<PropertyPath> string </PropertyPath>
(description)	Implements a data structure for describing a property as a path below another property, or below an owning type. Property paths are used in data binding to objects, and in storyboards and timelines for animations.
[types assignable to]	<a href="#">PropertyPath</a> <a href="#">x:Object</a>
(used by)	<a href="#">Binding</a> <a href="#">Storyboard</a>
[text syntax]	<a href="#">PropertyPathSyntax</a>
(properties)	
Path	<a href="#">x:String</a>
(description)	The path value held by this PropertyPath.

## 1.66 Rect

(usage)	<Rect> string </Rect>
(description)	Describes the width, height, and point origin of a rectangle.
[types assignable to]	<a href="#">Rect</a> <a href="#">x:Object</a>
(used by)	<a href="#">Rect</a> <a href="#">RectangleGeometry</a>
[is default constructible]	False
[is nullable]	False
[text syntax]	<a href="#">RectSyntax</a>
(properties)	
Height	<a href="#">x:Double</a>
(description)	The height of the rectangle.
Width	<a href="#">x:Double</a>

(description)	The width of the rectangle.
X	<a href="#">x:Double</a>
(description)	The x-axis value of the left side of the rectangle.
Y	<a href="#">x:Double</a>
(description)	The y-axis value of the top side of the rectangle.

## 1.67 RepeatBehavior

(usage)	<RepeatBehavior> string </RepeatBehavior>
(description)	Describes how a Timeline repeats its simple duration.
[types assignable to]	<a href="#">RepeatBehavior</a> <a href="#">x:Object</a>
(used by)	<a href="#">RepeatBehavior</a> <a href="#">Timeline</a>
[is default constructible]	False
[is nullable]	False
[text syntax]	<a href="#">RepeatBehaviorSyntax</a>

## 1.68 RoutedEvent

(usage)	<RoutedEvent> string </RoutedEvent>
(description)	Represents a routed event to the Silverlight event system.
[types assignable to]	<a href="#">RoutedEvent</a> <a href="#">x:Object</a>
(used by)	<a href="#">EventTrigger</a>
[text syntax]	<a href="#">RoutedEventSyntax</a>

## 1.69 RowDefinitionCollection

(usage)	None.
(description)	Provides access to an ordered, strongly typed collection of RowDefinition objects.



[types assignable to]	<a href="#">RowDefinitionCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">Grid</a>
[is list]	True
[allowed types]	<a href="#">RowDefinition</a>

## 1.70 ScrollBarVisibility

(usage)	<a href="#">Disabled</a>   <a href="#">Auto</a>   <a href="#">Hidden</a>   <a href="#">Visible</a>
(description)	Specifies the visibility of ScrollViewercontrol.
[types assignable to]	<a href="#">ScrollBarVisibility</a> <a href="#">x:Object</a>
(used by)	<a href="#">ScrollViewer</a> <a href="#">TextBox</a>
[is nullable]	False
[text syntax]	<a href="#">ScrollBarVisibilitySyntax</a>

## 1.71 SetterBaseCollection

(usage)	<SetterBaseCollection> <a href="#">SetterBase</a> * </SetterBaseCollection>
(description)	Represents a collection of objects that inherit from SetterBase.
[types assignable to]	<a href="#">SetterBaseCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">Style</a>
[is list]	True
[allowed types]	<a href="#">SetterBase</a>

## 1.72 x:Single

link to externally defined type	<a href="#">x:Single</a> , from <a href="#">[MS-XAML]</a>
---------------------------------	---

## 1.73 Size

(usage)	<Size> string </Size>
(description)	Describes the width and height of an object.
[types assignable to]	<a href="#">Size</a> <a href="#">x:Object</a>
(used by)	<a href="#">ArcSegment</a> <a href="#">Size</a>
[is default constructible]	False
[is nullable]	False
[text syntax]	<a href="#">SizeSyntax</a>
(properties)	
Height	<a href="#">x:Double</a>
(description)	The Height of this instance of Size.
Width	<a href="#">x:Double</a>
(description)	The Width of this instance of Size.

## 1.74 Stretch

(usage)	<a href="#">None</a>   <a href="#">Fill</a>   <a href="#">Uniform</a>   <a href="#">UniformToFill</a>
(description)	Describes how content is resized to fill its allocated space.
[types assignable to]	<a href="#">Stretch</a> <a href="#">x:Object</a>
(used by)	<a href="#">Image</a> <a href="#">MediaElement</a> <a href="#">Shape</a> <a href="#">TileBrush</a>
[is nullable]	False
[text syntax]	<a href="#">StretchSyntax</a>

## 1.75 x:String

link to externally defined type	<a href="#">x:String</a> , from <a href="#">[MS-XAML]</a>
---------------------------------	---

## 1.76 StrokeCollection

(usage)	<StrokeCollection> <a href="#">Stroke</a> * </StrokeCollection>
(description)	Represents a collection of Stroke objects.
[types assignable to]	<a href="#">StrokeCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">InkPresenter</a>
[is list]	True
[allowed types]	<a href="#">Stroke</a>

## 1.77 StyleSimulations

(usage)	<a href="#">None</a>   [ <a href="#">BoldSimulation</a>   <a href="#">ItalicSimulation</a>   <a href="#">BoldItalicSimulation</a> ]*
(description)	Describes the simulation style of a font.
[types assignable to]	<a href="#">StyleSimulations</a> <a href="#">x:Object</a>
(used by)	<a href="#">Glyphs</a>
[is nullable]	False
[text syntax]	<a href="#">StyleSimulationsSyntax</a>

## 1.78 StylusPoint

(usage)	<StylusPoint />
(description)	Represents a single point collected while the user is entering ink strokes with the stylus or mouse.
[types assignable to]	<a href="#">StylusPoint</a> <a href="#">x:Object</a>
(used by)	<a href="#">StylusPointCollection</a>
[is default constructible]	False
[is nullable]	False
(properties)	
PressureFactor	<a href="#">x:Single</a>
(description)	The pressure factor of the stylus on the screen.
X	<a href="#">x:Double</a>
(description)	The value for the x-coordinate of the StylusPoint.
Y	<a href="#">x:Double</a>

**(description)**

The value for the y-coordinate of the StylusPoint.

## 1.79 StylusPointCollection

**(usage)**<StylusPointCollection> [ [StylusPointCollection](#) | [StylusPoint](#) ]\*  
</StylusPointCollection>**(description)**

Represents a collection of related StylusPoint objects.

**[types assignable to]**[StylusPointCollection](#) [x:Object](#)**(used by)**[Stroke](#) [StylusPointCollection](#)**[is list]**

True

**[allowed types]**[StylusPointCollection](#) [StylusPoint](#)

## 1.80 SweepDirection

**(usage)**[Counterclockwise](#) | [Clockwise](#)**(description)**

Specifies the direction an elliptical arc is drawn.

**[types assignable to]**[SweepDirection](#) [x:Object](#)**(used by)**[ArcSegment](#)**[is nullable]**

False

**[text syntax]**[SweepDirectionSyntax](#)

## 1.81 SystemParameters

**(usage)**{x:Static [SystemParameters.StaticPropertyName](#)}**(description)**

Contains properties that you can use to query system settings.

**[types assignable to]**[SystemParameters](#) [x:Object](#)

## 1.82 TextAlignment

(usage)	<a href="#">Center</a>   <a href="#">Left</a>   <a href="#">Right</a>
(description)	Specifies whether text is centered, left-aligned, or right-aligned.
[types assignable to]	<a href="#">TextAlignment</a> <a href="#">x:Object</a>
(used by)	<a href="#">TextBlock</a> <a href="#">TextBox</a>
[is nullable]	False
[text syntax]	<a href="#">TextAlignmentSyntax</a>

## 1.83 TextDecorationCollection

(usage)	<code>&lt;TextDecorationCollection&gt; string &lt;/TextDecorationCollection&gt;</code>
(description)	Provides the value for the <code>TextDecorations</code> and <code>TextDecorations</code> properties.
[types assignable to]	<a href="#">TextDecorationCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">Inline</a> <a href="#">TextBlock</a> <a href="#">TextDecorations</a>
[text syntax]	<a href="#">TextDecorationCollectionSyntax</a>

## 1.84 TextDecorations

(usage)	<code>{x:Static TextDecorations.StaticPropertyName}</code>
(description)	Implements a set of predefined text decorations.
[types assignable to]	<a href="#">TextDecorations</a> <a href="#">x:Object</a>

## 1.85 TextWrapping

(usage)	<a href="#">Wrap</a>   <a href="#">NoWrap</a>
(description)	Specifies whether text wraps when it reaches the edge of its container.
[types assignable to]	<a href="#">TextWrapping</a> <a href="#">x:Object</a>
(used by)	<a href="#">TextBlock</a> <a href="#">TextBox</a>
[is nullable]	False

[text syntax]	<a href="#">TextWrappingSyntax</a>
---------------	------------------------------------

## 1.86 Thickness

(usage)	<Thickness> string </Thickness>
(description)	Describes the thickness of a frame around a rectangle. Four Double values describe the Left, Top, Right, and Bottom sides of the rectangle, respectively.
[types assignable to]	<a href="#">Thickness</a> <a href="#">x:Object</a>
(used by)	<a href="#">Border</a> <a href="#">Control</a> <a href="#">FrameworkElement</a> <a href="#">TextBlock</a>
[is default constructible]	False
[is nullable]	False
[text syntax]	<a href="#">ThicknessSyntax</a>
(properties)	
Bottom	<a href="#">x:Double</a>
(description)	The width, in pixels, of the lower side of the bounding rectangle.
Left	<a href="#">x:Double</a>
(description)	The width, in pixels, of the left side of the bounding rectangle.
Right	<a href="#">x:Double</a>
(description)	The width, in pixels, of the right side of the bounding rectangle.
Top	<a href="#">x:Double</a>
(description)	The width, in pixels, of the upper side of the bounding rectangle.

## 1.87 TimelineCollection

(usage)	<TimelineCollection> <a href="#">Timeline</a> * </TimelineCollection>
(description)	Represents a collection of Timeline objects.
[types assignable to]	<a href="#">TimelineCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">Storyboard</a>
[is list]	True
[allowed types]	<a href="#">Timeline</a>

## 1.88 TimelineMarkerCollection

(usage)	<TimelineMarkerCollection> <a href="#">TimelineMarker</a> * </TimelineMarkerCollection>
(description)	Represents a collection of TimelineMarker objects that can be individually accessed by index.
[types assignable to]	<a href="#">TimelineMarkerCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">MediaElement</a>
[is list]	True
[allowed types]	<a href="#">TimelineMarker</a>

## 1.89 x:TimeSpan

link to externally defined type	<a href="#">x:TimeSpan</a> , from <a href="#">[MS-XAML]</a>
---------------------------------	---

## 1.90 ToolTipService

(usage)	None.
(description)	Represents a service that provides static methods to display a tooltip.
[types assignable to]	<a href="#">ToolTipService</a> <a href="#">x:Object</a>
(attachable properties)	
ToolTipService.ToolTip	<a href="#">x:Object</a>
(description)	A tooltip to be attached to a control.
[target type]	<a href="#">DependencyObject</a>

## 1.91 TransformCollection

(usage)	<TransformCollection> <a href="#">Transform</a> * </TransformCollection>
(description)	Represents a collection of Transform objects that can be individually accessed

---

	by index.
[types assignable to]	<a href="#">TransformCollection x:Object</a>
(used by)	<a href="#">TransformGroup</a>
[is list]	True
[allowed types]	<a href="#">Transform</a>

---

## 1.92 TriggerActionCollection

(usage)	<TriggerActionCollection> <a href="#">TriggerAction</a> * </TriggerActionCollection>
(description)	Represents a collection of BeginStoryboard objects.
[types assignable to]	<a href="#">TriggerActionCollection x:Object</a>
(used by)	<a href="#">EventTrigger</a>
[is list]	True
[allowed types]	<a href="#">TriggerAction</a>

---

## 1.93 TriggerCollection

(usage)	None.
(description)	Represents a collection of EventTrigger objects.
[types assignable to]	<a href="#">TriggerCollection x:Object</a>
(used by)	<a href="#">FrameworkElement</a>
[is list]	True
[allowed types]	<a href="#">TriggerBase</a>

---

## 1.94 UIElementCollection

(usage)	None.
(description)	Represents an ordered collection of UIElement objects.

---



[types assignable to]	<a href="#">UIElementCollection</a> <a href="#">x:Object</a>
(used by)	<a href="#">Panel</a>
[is list]	True
[allowed types]	<a href="#">UIElement</a>

## 1.95 x:Uri

link to externally defined type	<a href="#">x:Uri</a> , from <a href="#">[MS-XAML]</a>
---------------------------------	--

## 1.96 VerticalAlignment

(usage)	<a href="#">Top</a>   <a href="#">Center</a>   <a href="#">Bottom</a>   <a href="#">Stretch</a>
(description)	Describes how a child element is vertically positioned or stretched within a parent's layout slot.
[types assignable to]	<a href="#">VerticalAlignment</a> <a href="#">x:Object</a>
(used by)	<a href="#">Control</a> <a href="#">FrameworkElement</a>
[is nullable]	False
[text syntax]	<a href="#">VerticalAlignmentSyntax</a>

## 1.97 Visibility

(usage)	<a href="#">Visible</a>   <a href="#">Collapsed</a>
(description)	Specifies the display state of an element.
[types assignable to]	<a href="#">Visibility</a> <a href="#">x:Object</a>
(used by)	<a href="#">ScrollViewer</a> <a href="#">UIElement</a>
[is nullable]	False
[text syntax]	<a href="#">VisibilitySyntax</a>

## 1.98 XmlLanguage

<b>(usage)</b>	None.
<b>(description)</b>	Represents culture information for use in XML and XAML markup.
<b>[types assignable to]</b>	<a href="#">XmlLanguage</a> <a href="#">x:Object</a>
<b>(used by)</b>	<a href="#">FrameworkElement</a> <a href="#">Inline</a>

## 2 IEnumerable

<b>(usage)</b>	None.
<b>(description)</b>	Exposes the enumerator, which supports a simple iteration over a non-generic collection.
<b>[types assignable to]</b>	<a href="#">IEnumerable</a>
<b>(used by)</b>	<a href="#">ItemsControl</a>

## 3 IList

<b>(usage)</b>	None.
<b>(description)</b>	Represents a non-generic collection of objects that can be individually accessed by index.
<b>[types assignable to]</b>	<a href="#">IList</a>
<b>(used by)</b>	<a href="#">VisualStateManager</a>
<b>[is list]</b>	True
<b>[allowed types]</b>	<a href="#">x:Object</a>

## 4 IValueConverter

<b>(usage)</b>	None.
<b>(description)</b>	Exposes methods that allow modifying the data as it passes through the binding engine.

[types assignable to]	<a href="#">IValueConverter</a>
(used by)	<a href="#">Binding</a>

## 5.1 CultureInfo

(usage)	None.
(description)	Provides information about a specific culture. The information includes the names for the culture, the writing system, the calendar used, the formatting used for numbers and dates, and the order of sorted strings.
[types assignable to]	<a href="#">CultureInfo</a> <a href="#">x:Object</a>
(used by)	<a href="#">Binding</a> <a href="#">CultureInfo</a>
[is default constructible]	False

## 5.2 DateTimeFormatInfo

(usage)	None.
(description)	Defines how DateTime values are formatted and displayed, depending on the culture.
[types assignable to]	<a href="#">DateTimeFormatInfo</a> <a href="#">x:Object</a>
(used by)	<a href="#">CultureInfo</a> <a href="#">DateTimeFormatInfo</a>
[is default constructible]	False

## 5.3 DictionaryOfStringStringOfString

(usage)	None.
(description)	
[types assignable to]	<a href="#">DictionaryOfStringStringOfString</a> <a href="#">x:Object</a>
(used by)	<a href="#">MediaElement</a>
[is default constructible]	False
[allowed types]	<a href="#">x:String</a>
[allowed key types]	<a href="#">x:String</a>

## 5.4 Int32[]

(usage)	None.
(description)	
[types assignable to]	<a href="#">Int32[]</a> <a href="#">x:Array</a> <a href="#">x:Object</a>
(used by)	<a href="#">NumberFormatInfo</a>
[is default constructible]	False
[is list]	True

## 5.5 IsolatedStorageSettings

(usage)	None.
(description)	Provides a Dictionary`2 that stores key-value pairs in isolated storage.
[types assignable to]	<a href="#">IsolatedStorageSettings</a> <a href="#">x:Object</a>
(used by)	<a href="#">IsolatedStorageSettings</a>
[is default constructible]	False
[allowed types]	<a href="#">x:Object</a>
[allowed key types]	<a href="#">x:String</a>

## 5.6 NullableOfBoolean

(usage)	None.
(description)	
[types assignable to]	<a href="#">NullableOfBoolean</a> <a href="#">x:Object</a>
(used by)	<a href="#">ToggleButton</a>
[is default constructible]	False
[text syntax]	<a href="#">BooleanSyntax</a>

## 5.7 NullableOfColor

(usage) None.

(description)

[types assignable to] [NullableOfColor x:Object](#)

(used by) [ColorAnimation](#)

[is default constructible] False

## 5.8 NullableOfDouble

(usage) None.

(description)

[types assignable to] [NullableOfDouble x:Object](#)

(used by) [DoubleAnimation](#)

[is default constructible] False

[text syntax] [DoubleSyntax](#)

## 5.9 NullableOfInt32

(usage) None.

(description)

[types assignable to] [NullableOfInt32 x:Object](#)

(used by) [MediaElement](#)

[is default constructible] False

[text syntax] [Int32Syntax](#)

## 5.10 NullableOfPoint

(usage) None.

---

(description)

[types assignable to] [NullableOfPoint x:Object](#)

(used by) [PointAnimation](#)

[is default constructible] False

---

## 5.11 NullableOfTimeSpan

(usage) None.

(description)

[types assignable to] [NullableOfTimeSpan x:Object](#)

(used by) [Timeline](#)

[is default constructible] False

[text syntax] [TimeSpanSyntax](#)

---

## 5.12 NumberFormatInfo

(usage) None.

(description) Defines how numeric values are formatted and displayed, depending on the culture.

[types assignable to] [NumberFormatInfo x:Object](#)

(used by) [CultureInfo](#) [NumberFormatInfo](#)

[is default constructible] False

---

## 5.13 String[]

(usage) None.

(description)

[types assignable to] [String\[\] x:Array x:Object](#)

(used by) [DateTimeFormatInfo](#)

---

[is default constructible]	False
[is list]	True

---

## 5.14 SynchronizationContext

(usage)	None.
(description)	Provides the basic functionality for propagating a synchronization context in various synchronization models.
(types assignable to)	<a href="#">SynchronizationContext</a> <a href="#">x:Object</a>
(used by)	<a href="#">SynchronizationContext</a>
[is default constructible]	False

## Silverlight Xaml Text Syntax Information Sets

---

### AlignmentXSyntax

[values]

<b>Center</b>	The contents align toward the center of the container.
<b>Left</b>	The contents align toward the left of the container.
<b>Right</b>	The contents align toward the right of the container.

---

---

### AlignmentYSyntax

[values]

<b>Bottom</b>	The contents align toward the lower edge of the container.
<b>Center</b>	The contents align toward the center of the container.
<b>Top</b>	The contents align toward the upper edge of the container.

---

---

### BindingModeSyntax

[values]

<b>OneTime</b>	Updates the target property when the binding is created.
<b>OneWay</b>	Updates the target property when the binding is created. Changes to the source object can also propagate to the target.
<b>TwoWay</b>	Updates either the target or the source object when either changes. When the binding is created, the target property is updated from the source.

---

---

### BrushMappingModeSyntax

[values]

<b>Absolute</b>	The coordinate system is not relative to a bounding box. Values are interpreted
-----------------	---

---



directly in local space.

### RelativeToBoundingBox

The coordinate system is relative to a bounding box: 0 indicates 0 percent of the bounding box, and 1 indicates 100 percent of the bounding box. For example, (0.5, 0.5) describes a point in the middle of the bounding box, and (1, 1) describes a point at the bottom right of the bounding box.

## BrushSyntax

[values]

DRAFT NOTE - details coming in next version.

## ClickModeSyntax

[values]

### Hover

Specifies that the Click event should be raised when the mouse pointer moves over the control.

### Press

Specifies that the Click event should be raised when the mouse button is pressed and the mouse pointer is over the control. If you are using the keyboard, specifies that the Click event should be raised when the SPACEBAR or ENTER is pressed and the control has keyboard focus.

### Release

Specifies that the Click event should be raised when the left mouse button is pressed and released, and the mouse pointer is over the control. If you are using the keyboard, specifies that the Click event should be raised when the SPACEBAR or ENTER key is pressed and released, and the control has keyboard focus.

## ColorInterpolationModeSyntax

[values]

### ScRgbLinearInterpolation

Colors are interpolated in the sRGB color space

### SRgbLinearInterpolation

Colors are interpolated in the sRGB color space

---

## ColorSyntax

[values]

DRAFT NOTE - details coming in next version.

---

---

## CornerRadiusSyntax

[values]

DRAFT NOTE - details coming in next version.

---

---

## CultureInfoletfLanguageTagSyntax

[patterns]

[A-Z]{2}(-[0-9A-Z]{2})

A two letter language code identifier optionally followed by a dash then a two-letter country or region code.

---

---

## CursorSyntax

[values]

DRAFT NOTE - details coming in next version.

---

---

## DeepZoomImageTileSourceSyntax

[values]

DRAFT NOTE - details coming in next version.

---

---

## DoubleCollectionSyntax

[values]

DRAFT NOTE - details coming in next version.

---

---

## DurationSyntax

[values]

DRAFT NOTE - details coming in next version.

---

## FillBehaviorSyntax

[values]

**HoldEnd**

After it reaches the end of its active period, the timeline holds its progress until the end of its parent's active and hold periods.

**Stop**

The timeline stops if it is outside its active period while its parent is inside its active period.

---

## FillRuleSyntax

[values]

**EvenOdd**

Rule that determines whether a point is in the fill region by drawing a ray from that point to infinity in any direction and counting the number of path segments within the given shape that the ray crosses. If this number is odd, the point is inside; if even, the point is outside.

**Nonzero**

Rule that determines whether a point is in the fill region of the path by drawing a ray from that point to infinity in any direction and then examining the places where a segment of the shape crosses the ray. Starting with a count of 0, add one each time a segment crosses the ray from left to right and subtract one each time a path segment crosses the ray from right to left. After counting the crossings, if the result is 0 then the point is outside the path. Otherwise, it is inside.

---

## FontFamilySyntax

[values]

DRAFT NOTE - details coming in next version.

---

## FontStretchSyntax

---

---

[values]

DRAFT NOTE - details coming in next version.

---

## FontStyleSyntax

---

[values]

DRAFT NOTE - details coming in next version.

---

## FontWeightSyntax

---

[values]

DRAFT NOTE - details coming in next version.

---

## GeometrySyntax

---

[values]

DRAFT NOTE - details coming in next version.

---

## GradientSpreadMethodSyntax

---

[values]

<b>Pad</b>	The color values at the ends of the gradient vector fill the remaining space.
<b>Reflect</b>	The gradient is repeated in the reverse direction until the space is filled.
<b>Repeat</b>	The gradient is repeated in the original direction until the space is filled.

---

## GridLengthSyntax

---

[values]

DRAFT NOTE - details coming in next version.

---

---

## HorizontalAlignmentSyntax

[values]

<b>Center</b>	An element aligned to the center of the layout slot for the parent element.
<b>Left</b>	An element aligned to the left of the layout slot for the parent element.
<b>Right</b>	An element aligned to the right of the layout slot for the parent element.
<b>Stretch</b>	An element stretched to fill the entire layout slot of the parent element.

---

## ImageSourceSyntax

[values]

DRAFT NOTE - details coming in next version.

---

## KeyboardNavigationModeSyntax

[values]

<b>Cycle</b>	Focus returns to the first or the last keyboard navigation stop inside of a container when the first or last keyboard navigation stop is reached.
<b>Local</b>	Tab indexes are considered on the local subtree only inside this container.
<b>Once</b>	The container and all of its child elements as a whole receive focus only once.

---

## KeySplineSyntax

[values]

DRAFT NOTE - details coming in next version.

---

## KeyTimeSyntax

[values]

DRAFT NOTE - details coming in next version.

## LineStackingStrategySyntax

[values]

<b>BlockLineHeight</b>	The stack height is determined by the block element line-height property value.
<b>MaxHeight</b>	The stack height is the smallest value that contains the extended block progression dimension of all the inline elements on that line when those elements are properly aligned.

## MatrixSyntax

[values]

DRAFT NOTE - details coming in next version.

## ModifierKeysSyntax

[values]

<b>Alt</b>	The ALT key is pressed.
<b>Apple</b>	The Apple key (also known as the Open Apple key) is pressed.
<b>Control</b>	The CTRL key is pressed.
<b>None</b>	No modifiers are pressed.
<b>Shift</b>	The SHIFT key is pressed.
<b>Windows</b>	The Windows logo key is pressed.

## OrientationSyntax

[values]

<b>Horizontal</b>	The control or layout should be horizontally oriented.
<b>Vertical</b>	The control or layout should be vertically oriented.

## PenLineCapSyntax

[values]

<b>Flat</b>	A cap that does not extend past the last point of the line. Comparable to no line cap.
<b>Round</b>	A semicircle that has a diameter equal to the line thickness.
<b>Square</b>	A rectangle that has a height equal to the line thickness and a length equal to half the line thickness.
<b>Triangle</b>	An isosceles right triangle whose base length is equal to the thickness of the line.

## PenLineJoinSyntax

[values]

<b>Bevel</b>	Line joins use beveled vertices.
<b>Miter</b>	Line joins use regular angular vertices.
<b>Round</b>	Line joins use rounded vertices.

## PointCollectionSyntax

[values]

DRAFT NOTE - details coming in next version.

## PointSyntax

[values]

DRAFT NOTE - details coming in next version.

## PropertyPathSyntax

[values]

DRAFT NOTE - details coming in next version.

---

## RectSyntax

[values]

DRAFT NOTE - details coming in next version.

---

## RepeatBehaviorSyntax

[values]

DRAFT NOTE - details coming in next version.

---

## RoutedEventSyntax

[values]

DRAFT NOTE - details coming in next version.

---

## ScrollBarVisibilitySyntax

[values]

**Auto**

A ScrollBar appears and the dimension of the ScrollViewer is applied to the content when the viewport cannot display all of the content. For a horizontal ScrollBar, the width of the content is set to the ViewportWidth of the ScrollViewer. For a vertical ScrollBar, the height of the content is set to the ViewportHeight of the ScrollViewer.

**Disabled**

A ScrollBar does not appear even when the viewport cannot display all of the content. The dimension of the content is set to the corresponding dimension of the ScrollViewer parent. For a horizontal ScrollBar, the width of the content is set to the ViewportWidth of the ScrollViewer. For a vertical ScrollBar, the height of the content is set to the ViewportHeight of the ScrollViewer.

**Hidden**

A ScrollBar does not appear even when the viewport cannot display all of the content. The dimension of the ScrollViewer is not applied to the content.

**Visible**

A ScrollBar always appears. The dimension of the ScrollViewer is applied to the content. For a horizontal ScrollBar, the width of the content is set to the ViewportWidth of the ScrollViewer. For a vertical ScrollBar, the height of the content is set to the ViewportHeight of the ScrollViewer.

---



## SizeSyntax

[values]

DRAFT NOTE - details coming in next version.

## StretchSyntax

[values]

<b>Fill</b>	The content is resized to fill the destination dimensions. The aspect ratio is not preserved.
<b>None</b>	The content preserves its original size.
<b>Uniform</b>	The content is resized to fit in the destination dimensions while it preserves its native aspect ratio.
<b>UniformToFill</b>	The content is resized to fill the destination dimensions while it preserves its native aspect ratio. If the aspect ratio of the destination rectangle differs from the source, the source content is clipped to fit in the destination dimensions.

## StyleSimulationsSyntax

[values]

<b>BoldItalicSimulation</b>	Bold and Italic style simulation.
<b>BoldSimulation</b>	Bold style simulation.
<b>ItalicSimulation</b>	Italic style simulation.
<b>None</b>	No font style simulation.

## SweepDirectionSyntax

[values]

<b>Clockwise</b>	Arcs are drawn in a clockwise (positive-angle) direction.
<b>Counterclockwise</b>	Arcs are drawn in a counterclockwise (negative-angle) direction.

---

## TextAlignmentSyntax

---

### [values]

<b>Center</b>	Text is centered.
<b>Left</b>	Text is aligned to the left.
<b>Right</b>	Text is aligned to the right.

---

---

## TextDecorationCollectionSyntax

---

### [values]

DRAFT NOTE - details coming in next version.

---

---

## TextWrappingSyntax

---

### [values]

<b>NoWrap</b>	No line wrapping is performed.
<b>Wrap</b>	Line breaking occurs if a line of text overflows beyond the available width of its container. Line breaking occurs even if the standard line-breaking algorithm cannot determine any line break opportunity, such as when a line of text includes a long word that is constrained by a fixed-width container without scrolling.

---

---

## ThicknessSyntax

---

### [values]

DRAFT NOTE - details coming in next version.

---

---

## TimeSpanSyntax

---

### [patterns]

---

- `?(\d+\.)?\d\d?:\d\d?:((\d\d?)|(\d?\d?\.\d*))`

The string may optionally begin with a decimal number and a dot. When present, this optional part indicates the number of days. The string always contains three parts separated by colons. The first two of these are one or two digit numbers specifying hours and minutes. The third part indicates the number of seconds and can be a two-digit integer, or a floating point number with 0, 1, or 2 digits before the point, and any number of digits after the point. Examples:

0:0:1 One second

1.0:0:0 One day

0:0:0.5 Half a second

0:0:.5 Half a second

2.5:2:22 Two days, five hours, two minutes, 22 seconds

[is case sensitive]

true

-?\d+

A decimal number specifying the number of days. Examples:

1 One day

10 Ten days

[is case sensitive]

true

## TransformSyntax

[values]

DRAFT NOTE - details coming in next version.

## VerticalAlignmentSyntax

[values]

**Bottom**

The element is aligned to the bottom of the parent's layout slot.

**Center**

The element is aligned to the center of the parent's layout slot.

**Stretch**

The element is stretched to fill the entire layout slot of the parent element.

**Top**

The element is aligned to the top of the parent's layout slot.

---

## VisibilitySyntax

---

[values]

**Collapsed**

Do not display the element, and do not reserve space for it in layout.

**Visible**

Display the element.

---

## Appendix A: References

### [MS-XAML]

*Xaml Object Mapping Specification 2006*, Microsoft. 30 June 2008. <http://go.microsoft.com/fwlink/?LinkId=114525>

### [MS-WPFXV]

*WPF Xaml Vocabulary Specification 2006*, Microsoft. 30 June 2008. <http://go.microsoft.com/fwlink/?LinkId=114526>

## Appendix B: Microsoft Silverlight Behavior

The information in this specification is applicable to the following versions of the Microsoft product:

- Silverlight 2

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Silverlight behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Silverlight does not follow the prescription.

## Index

AlignmentX.....	14	DateTimeFormatInfo .....	115
AlignmentY.....	14	DeepZoomImageTileSource.....	41
Application .....	14	DependencyObject .....	22
ArcSegment.....	43	DependencyProperty.....	90
Array.....	15	Deployment .....	28
AssemblyPart .....	22	DictionaryOfStringStringOfString.....	115
AssemblyPartCollection .....	15	DiscreteColorKeyFrame .....	27
AutomationProperties .....	15	DiscreteDoubleKeyFrame .....	29
BeginStoryboard .....	54	DiscreteObjectKeyFrame .....	42
BezierSegment .....	43	DiscretePointKeyFrame .....	46
Binding .....	17	Double .....	90
BindingMode.....	17	DoubleAnimation.....	51
BitmapImage .....	38	DoubleAnimationUsingKeyFrames .....	51
Boolean .....	18	DoubleCollection .....	90
Border .....	59	DoubleKeyFrame .....	28
Brush .....	23	DoubleKeyFrameCollection .....	91
BrushMappingMode .....	18	DrawingAttributes .....	30
Button .....	63	Duration.....	91
ButtonBase.....	62	Ellipse.....	85
Byte .....	18	EllipseGeometry.....	35
Canvas .....	80	EventTrigger.....	55
Char .....	18	FillBehavior .....	91
CheckBox.....	65	FillRule .....	92
ClickMode .....	18	FontFamily .....	92
Color.....	19	FontSource.....	92
ColorAnimation .....	50	FontStretch .....	92
ColorAnimationUsingKeyFrames .....	51	FontStretches.....	93
ColorInterpolationMode .....	19	FontStyle .....	93
ColorKeyFrame.....	26	FontStyles .....	93
ColorKeyFrameCollection.....	20	FontWeight.....	94
Colors .....	20	FontWeights .....	94
ColumnDefinition .....	27	FrameworkElement .....	57
ColumnDefinitionCollection .....	20	FrameworkTemplate .....	30
ComboBox.....	70	GeneralTransform.....	31
ComboBoxItem .....	66	Geometry.....	35
ContentControl .....	62	GeometryCollection.....	94
ContentPresenter.....	59	GeometryGroup.....	35
Control .....	60	Glyphs .....	76
ControlTemplate .....	30	GradientBrush.....	23
CornerRadius.....	21	GradientSpreadMethod.....	95
CultureInfo .....	115	GradientStop.....	37
Cursor .....	21	GradientStopCollection .....	95
Cursors .....	22	Grid .....	81
DataTemplate.....	31	GridLength .....	95

HorizontalAlignment	96	NullableOfPoint	117
HyperlinkButton	63	NullableOfTimeSpan	118
IEnumerable	114	NumberFormatInfo	118
IList	114	Object	14
Image	77	ObjectAnimationUsingKeyFrames	52
ImageBrush	25	ObjectKeyFrame	41
ImageSource	38	ObjectKeyFrameCollection	99
InkPresenter	81	Orientation	100
Inline	38	Panel	80
InlineCollection	96	PasswordBox	71
InputMethod	40	Path	85
Int32	96	PathFigure	42
Int32[]	116	PathFigureCollection	100
IsolatedStorageSettings	116	PathGeometry	36
ItemCollection	96	PathSegment	43
ItemsControl	68	PathSegmentCollection	100
ItemsPanelTemplate	31	PenLineCap	101
ItemsPresenter	77	PenLineJoin	101
IValueConverter	114	Point	101
Keyboard	97	PointAnimation	52
KeyboardNavigationMode	97	PointAnimationUsingKeyFrames	53
KeySpline	40	PointCollection	102
KeyTime	97	PointKeyFrame	46
LicenseAcquirer	98	PointKeyFrameCollection	102
Line	85	PolyBezierSegment	44
LinearColorKeyFrame	27	Polygon	86
LinearDoubleKeyFrame	29	Polyline	86
LinearGradientBrush	24	PolyLineSegment	44
LinearPointKeyFrame	46	PolyQuadraticBezierSegment	45
LineBreak	39	Popup	83
LineGeometry	36	PresentationFrameworkCollectionOfT	47
LineSegment	44	ProgressBar	72
LineStackingStrategy	98	PropertyPath	103
ListBox	70	QuadraticBezierSegment	45
ListBoxItem	65	RadialGradientBrush	24
Matrix	98	RadioButton	65
MatrixTransform	32	RangeBase	72
MediaElement	77	Rect	103
ModifierKeys	99	Rectangle	87
MultiScaleImage	79	RectangleGeometry	37
MultiScaleTileSource	41	RepeatBehavior	104
NullableOfBoolean	116	RepeatButton	64
NullableOfColor	117	ResourceDictionary	47
NullableOfDouble	117	RotateTransform	32
NullableOfInt32	117	RoutedEvent	104



RowDefinition .....	47	TextBlock .....	87
RowDefinitionCollection .....	104	TextBox .....	74
Run .....	39	TextDecorationCollection .....	109
ScaleTransform .....	33	TextDecorations.....	109
ScrollBar .....	73	TextWrapping .....	109
ScrollBarVisibility.....	105	Thickness .....	110
ScrollContentPresenter.....	60	Thumb.....	75
ScrollViewer .....	66	TileBrush .....	25
Selector .....	69	Timeline .....	49
Setter.....	48	TimelineCollection .....	110
SetterBase .....	48	TimelineMarker .....	54
SetterBaseCollection.....	105	TimelineMarkerCollection .....	111
Shape.....	83	TimeSpan .....	111
Single.....	105	ToggleButton .....	64
Size .....	105	ToolTip .....	68
SkewTransform .....	33	ToolTipService.....	111
Slider .....	73	Transform .....	32
SolidColorBrush.....	25	TransformCollection .....	111
SplineColorKeyFrame.....	27	TransformGroup .....	34
SplineDoubleKeyFrame.....	29	TranslateTransform .....	34
SplinePointKeyFrame.....	46	TriggerAction .....	54
StackPanel .....	82	TriggerActionCollection .....	112
Storyboard .....	53	TriggerBase .....	55
Stretch.....	106	TriggerCollection.....	112
String .....	106	UIElement .....	56
String[].....	118	UIElementCollection .....	112
Stroke .....	48	Uri .....	113
StrokeCollection.....	106	UserControl.....	76
Style.....	49	VerticalAlignment.....	113
StyleSimulations.....	107	VideoBrush .....	26
StylusPoint .....	107	Visibility.....	113
StylusPointCollection .....	108	VisualState .....	88
SweepDirection.....	108	VisualStateGroup.....	89
SynchronizationContext.....	119	VisualStateManager.....	89
SystemParameters .....	108	XamlType .....	99
TextAlignment.....	109	XmlLanguage.....	114