Sitecore® Experience Platform™
SPEAK Component Reference

Table of Contents

Chapter 1

Introduction

This chapter briefly describes how the SPEAK components are organized.

1.1 Organization of SPEAK Components

The SPEAK components are stored in two different places in the Sitecore content tree:

- Core components are in sitecore/client/Speak/Layouts/Renderings, in these subfolders:
  - Common
  - Data
  - Resources

- Components for `Sitecore.Services.Client` are in the `Sitecore/client/Services/Layouts/Renderings` folder.

- The rest of the components are stored in `sitecore/client/Business Component Library/Layouts/Renderings`, in these subfolders:
  - Authentications
  - Behaviors
  - Charts
  - Common
  - Containers
  - Dialogs
  - Forms
  - Indicators
  - Lists and Grids
  - Navigation
  - Searching and Filtering
  - Silverlight
  - Structures

Two of these subfolders have the name, "Common." The **Add Rendering** dialog box merges the components from these subfolders.

1.2 Changes in this version

Accordion	Added ImageUrl property.
ActionControl	Additional Model and View Members.
ActionControl	Favorites are now stored for the current page in each user profile.
ActionControl	Spelling errors in some model member names corrected.
Behavior property	All descriptions of the Behavior property changed to say "space-separated" instead of "comma-separated".
Border	Height property added.
Chart- and GenericDataProviders	Descriptions now mentions getData().
ComboBox	Changes in properties; new rebind function.
DatePicker	Many changes.
EntityDataSource	Added in the Services group.
Image	Changed list of model members.
ListBox	Changes in properties; new rebind function.
ListControl	Changed description of the Sorting property; added "sorting" model member.
ListControl	New property: AllowMultipleColumnSorting.
MultiSelectList behavior	Added check/uncheck methods to the description of the MultiSelectList behavior.
ProtectedBorder	Component added.
QueryDataSource	Formatting property added.
Repeater	Clarified description.
Services	A new folder where the data sources that use Sitecore.Services.Client are located.
SilverlightFrame	Component added.
SmartPanel	Dimension property renamed to Dimensions.
StoredQueryDataSource	Added in the Service group.
Tooltip	Animation property removed.
TreeView	Added ShowHiddenItems property.
TreeView	Expanded description of the ShowHidden property.
Uploader and UploaderInfo	New properties: BrowseButtonText, ShortDescriptionText; ShowAlternateTextField, ShowDescriptionField, ShowNameField.
Uploader	DestinationUrl property added.

Chapter 2

Authentications

2.1 AccountInformation

The AccountInformation control shows two things:

- The name of the user that is logged in
- A "logout" link

When users click the link, Sitecore logs them out.

2.1.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control is visible.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False

VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Database	Specify the database that the Uploader control uploads files to: • The exact name of a database, for example "core", or "master". • $context_database: context database. • $context_contentdatabase: content database. The control places the files in the Media Library of the database you specify.	The context database.
Design		
Id	Lets you set the ID of the control.	

2.1.2   Model Members

There are no model members for this control.

2.1.3   View Members

There are no view members for this control.

2.1.4   Configuration

There are no configuration item templates for this control.

Chapter 3

Behaviors

3.1 MultiSelectList

The MultiSelectList is only a relevant behavior for the ListControl control.

The MultiSelectList behavior extends the ListControl with two properties:

- checkedItems - contains an array of the raw objects that the user has checked.

- checkedItemIds - contains an array of the IDs of the items that the user has checked.

3.1.1 Methods

The MultiSelect behavior also extends the ListControl with eight methods. You use these methods to control how and when items are checked and unchecked when the default functionality is not enough for your requirements.

The methods and their arguments are:

- checkItem(item)

- checkItem(itemId)

- checkItems(array of items)

- checkItems(array of itemIds)

- uncheckItem(item)

- uncheckItem(itemId)

- uncheckItems(array of items)

- uncheckItems(array of itemIds)

Usage*:*

```
ListControl.viewModel.checkItem("{EB443C0B-F923-409E-85F3-E7893C8C30C2}")
```

When you call the `checkItem()` method like this, the MultiSelectList behavior

1. adds the ID of the specified item to the checkedItemIds array, and

2. if ListControl contains a row for the specified item, then it adds this item to the checkedItems array and marks the row as checked.

The MultiSelectList behavior scans the checkedItemIds array when a new row is added to the ListControl, and it marks the appropriate rows as checked. It also updates the checkedItems array.

This means that the checkedItemIds array contains ID of items that have been requested to be checked, and the checkedItems array refers to items that are actually checked and shown in the ListControl.

3.2   Scrollbar

The Scrollbar is only a relevant behavior for the ListControl control. The Scrollbar behavior adds a scrollbar to ListControl.

Chapter 4

Charts

4.1 Error Handling in Chart Components, in Data Providers and in the Value Label

The chart components, the data providers, and the value label can all trigger an error event. When they trigger this event, they pass an object to the event. You can handle the error in different ways: you can use a MessageBar control, for example, or you can write to a console log.

This PageCode JavaScript example writes an error object from an AreaChart control to a console log:

```
this.AreaChart.on("error", function (errorObject) {
  console.log(errorObject);
});
```

This `errorObject` contains two mandatory properties, `name` and `message`, and an optional property called `response`.

- Name

  This property identifies the kind of error, for example "info", "warning", or "error".

- Message

  This is typically the error message itself.

- Response

  If the error is spawned by an AJAX call to a DataProvider, this is the response.

4.2 AreaChart

The AreaChart control shows an area chart. An area chart is very similar to a line chart. The difference is that in an area chart, the control fills the area below the line that connects the data points with color.

The AreaChart control supports multiseries charts. When an AreaChart control shows a multiseries chart, you can specify if it shows the series stacked or unstacked.

The AreaChart control supports showing a sequence of charts when the data it receives from the data provider contains multiple data sets.

4.2.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
ShowAxis	If true, the control shows axes.	True
ShowLegend	If true, the control shows a legend for the chart.	True
ShowValues	If true, the control shows the values of the data points inside the graph itself.	False
SingleChartHeight	Specify the height in pixels of the chart. If the chart is part of sequence of charts, the control gives each individual chart this height.	400
SingleChartWidth	If you specify "100%, the chart takes up the available width. If you specify a number, the control gives the chart this width in pixels. If the chart is part of sequence of charts, "100%" means each chart has the entire available width, and the chart sequence becomes a vertical sequence. If the chart is part of sequence of charts, and you specify an integer, the	100%

	control gives each chart this width in pixels, and the sequence of charts wraps horizontally.	
StackSeries	If true, the control stacks the distinctive series in a multiseries chart.	False
Tooltip	Specify the text that you want displayed when the user hovers the mouse over the chart.	None
VisibleCategoriesRange	If a multi-series chart and this number is > 0, the control will show this number of categories along the horizontal axis. If the data contains more categories, users have to scroll horizontally. If this number is 0, the control shows all categories.	0
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource		
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source	False

	of the control.	
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.DataProvider.Data}	
DataMapping		
CategoryChartField	Point to a ChartField item. The control uses this field as the X (horizontal) axis of the chart.	
CategoryFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values to filter the data column the CategoryChartField points to.	
SeriesChartField	Point to a ChartField. The control uses this field as the definition of the series it shows.	
SeriesFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values as a facet of the data column the SeriesChartField points to. If you do not enter anything, the control will create a facet for each distinct value in the data column.	

ValueChartFields	Point to a ChartField item. The control uses this field as the Y (vertical) axis of the chart.	
Design		
Id	Lets you set the ID of the control.	

4.2.2 Configuration

You can create a configuration item for an AreaChart by creating an item based on the AreaChart Parameters template.

The configuration item has these inherited fields:

- ShowAxis

- ValueChartFields

- SingleChartWidth

- CategoryChartField

- ShowLegend

- Data

- ShowValues

- CategoryFilter

- SingleChartHeight

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

- VisibleCategoriesRange

- SeriesChartField

- SeriesFilter

- StackSeries

4.3 BarChart

The BarChart control shows a bar chart. A bar chart shows rectangular bars with lengths proportional to the values that they represent. The control plots the bars horizontally. The ColumnChart control is very similar, except that this control plots the bars ("columns") vertically.

The BarChart supports multiseries charts. When a BarChart control shows a multiseries chart, you can decide if it shows the series stacked or unstacked.

The BarChart control supports showing a sequence of charts when the data it receives from the data provider contains multiple data sets.

4.3.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
ShowAxis	If true, the control shows axes.	True
ShowLegend	If true, the control shows a legend for the chart.	True
ShowValues	If true, the control shows the values of the data points inside the graph itself.	False
SingleChartHeight	Specify the height in pixels of the chart. If the chart is part of sequence of charts, the control gives each individual chart this height.	400
SingleChartWidth	If you specify "100%", the chart takes up the available width. If you specify a number, the control gives the chart this width in pixels. If the chart is part of sequence of charts, "100%" means each chart has the entire available width, and the chart sequence becomes a vertical sequence. If the chart is part of sequence of charts, and	100%

	you specify an integer, the control gives each chart this width in pixels, and the sequence of charts wraps horizontally.	
StackSeries	If true, the control stacks the distinctive series in a multiseries chart.	False
Tooltip	Specify the text that you want displayed when the user hovers the mouse over the chart.	None
VisibleCategoriesRange	If a multi-series chart and this number is > 0, the control will show this number of categories along the horizontal axis. If the data contains more categories, users have to scroll horizontally. If this number is 0, the control shows all categories.	0
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource		
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies	False

	based on the data source of the control.	
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.DataProvider.Data}	
DataMapping		
CategoryChartField	Point to a ChartField item. The control uses this field as the X (horizontal) axis of the chart.	
CategoryFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values to filter the data column the CategoryChartField points to.	
SeriesChartField	Point to a ChartField. The control uses this field as the definition of the series it shows.	
SeriesFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values as a facet of the data column the SeriesChartField points to. If you do not enter anything, the control will create a facet for each distinct value	

	in the data column.	
ValueChartFields	Point to a ChartField item. The control uses this field as the Y (vertical) axis of the chart.	
Design		
Id	Lets you set the ID of the control.	

4.3.2 Configuration

You can create a configuration item for a BarChart by creating an item based on the BarChart Parameters template.

The configuration item has these inherited fields:

- ShowAxis

- ValueChartFields

- SingleChartWidth

- CategoryChartField

- ShowLegend

- Data

- ShowValues

- CategoryFilter

- SingleChartHeight

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

- VisibleCategoriesRange

- SeriesChartField

- SeriesFilter

- StackSeries

4.4 ColumnChart

The ColumnChart control shows a column chart. A colums chart shows rectangular columns with lengths proportional to the values that they represent. The control plots the columns vertically. The BarChart control is very similar, except that this control plots the columns ("bars") horizontally.

The ColumnChart supports multiseries charts. When a ColumnChart control shows a multiseries chart, you can decide if it shows the series stacked or unstacked.

The ColumnChart control supports showing a sequence of charts when the data it receives from the data provider contains multiple data sets.

4.4.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
ShowAxis	If true, the control shows axes.	True
ShowLegend	If true, the control shows a legend for the chart.	True
ShowValues	If true, the control shows the values of the data points inside the graph itself.	False
SingleChartHeight	Specify the height in pixels of the chart. If the chart is part of sequence of charts, the control gives each individual chart this height.	400
SingleChartWidth	If you specify "100%, the chart takes up the available width. If you specify a number, the control gives the chart this width in pixels. If the chart is part of sequence of charts, "100%" means each chart has the entire available width, and the chart sequence becomes a vertical sequence. If the chart is part of sequence of charts, and	100%

	you specify an integer, the control gives each chart this width in pixels, and the sequence of charts wraps horizontally.	
StackSeries	If true, the control stacks the distinctive series in a multiseries chart.	False
Tooltip	Specify the text that you want displayed when the user hovers the mouse over the chart.	None
VisibleCategoriesRange	If a multi-series chart and this number is > 0, the control will show this number of categories along the horizontal axis. If the data contains more categories, users have to scroll horizontally. If this number is 0, the control shows all categories.	0
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource		
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies	False

	based on the data source of the control.	
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.DataProvider.Data}	
DataMapping		
CategoryChartField	Point to a ChartField item. The control uses this field as the X (horizontal) axis of the chart.	
CategoryFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values to filter the data column the CategoryChartField points to.	
SeriesChartField	Point to a ChartField. The control uses this field as the definition of the series it shows.	
SeriesFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values as a facet of the data column the SeriesChartField points to. If you do not enter anything, the control will create a facet for each distinct value	

	in the data column.	
ValueChartFields	Point to a ChartField item. The control uses this field as the Y (vertical) axis of the chart.	
Design		
Id	Lets you set the ID of the control.	

4.4.2 Configuration

You can create a configuration item for a ColumnChart by creating an item based on the ColumnChart Parameters template.

The configuration item has these inherited fields:

- ShowAxis
- ValueChartFields
- SingleChartWidth
- CategoryChartField
- ShowLegend
- Data
- ShowValues
- CategoryFilter
- SingleChartHeight
- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id
- VisibleCategoriesRange
- SeriesChartField
- SeriesFilter
- StackSeries

4.5 DoughnutChart

The DoughnutChart control shows a doughnut chart. A doughnut chart displays values as percentages of a whole, and each value is an individual slice of the whole. Doughnut charts are functionally identical to pie charts.

The DoughnutChart control does not support multiseries charts.

The DoughnutChart control supports showing a sequence of charts when the data it receives from the data provider contains multiple data sets.

4.5.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
ShowLegend	If true, the control shows a legend for the chart.	True
ShowValues	If true, the control shows the values of the data points inside the graph itself.	False
SingleChartHeight	Specify the height in pixels of the chart. If the chart is part of sequence of charts, the control gives each individual chart this height.	400
SingleChartWidth	If you specify "100%, the chart takes up the available width. If you specify a number, the control gives the chart this width in pixels. If the chart is part of sequence of charts, "100%" means each chart has the entire available width, and the chart sequence becomes a vertical sequence. If the chart is part of sequence of charts, and you specify an integer, the control gives each chart this width in pixels, and the	100%

	sequence of charts wraps horizontally.	
Tooltip	Specify the text that you want displayed when the user hovers the mouse over the chart.	None
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource		
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether	False

	output caching varies based on query string parameters passed in the URL.	
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.DataProvider.Data}	
DataMapping		
CategoryChartField	Point to a ChartField item. The control uses this field labels for the slices.	
CategoryFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values to filter the data column the CategoryChartField points to.	
ValueChartFields	Point to a ChartField item. The control uses this field as values of the slices.	
Design		
Id	Lets you set the ID of the control.	

4.5.2 Configuration

You can create a configuration item for a DoughnutChart by creating an item based on the DoughnutChart Parameters template.

The configuration item has these inherited fields:

- ShowAxis

- ValueChartFields

- SingleChartWidth

- CategoryChartField

- ShowLegend

- Data

- ShowValues

- CategoryFilter

- SingleChartHeight

- IsVisible

- AccessKey

- Tooltip
- Behaviors
- Id

4.6   JourneyChart

The JourneyChart control shows an extended ColumnChart chart. The JourneyChart control adds the following features to the ColumnChart:

- Users can select the time range ("duration") of the data the control shows in the chart.

- Users can page back and forth. The control checks if there is data available for date ranges before or after the one that it currently shows. If so, the control shows page back and page forward buttons as appropriate.

- You can specify that users can "drill up" and "drill down" through time ranges.

- You can specify that users can open a Details Panel for the segments of the chart.

4.6.1   Duration and Units

The *Duration* and *Unit* concepts are important for understanding how the JourneyChart control works. The Duration is the time range of the data the control displays. A Unit is the time range of a single segment of the chart.

You create durations by creating Duration configuration items. You put the items you create in a folder under PageSettings, and you specify this folder in the DurationItems property of the Journey control

You create a Duration item by creating an item based on the JourneyDuration template.

The item has these fields:

Field	Description
Duration	Specify the time range of the data the control shows. The possible values: Day, Week, Month, Quarter, and Year.
Unit	Specify the time range of single segment of the chart. The possible values: Hour, Day, Week, Month, Quarter, and Year.
UnitStep	Specify how many units the control displays at a time (in a "page").
CanDrillUp	If true, the control shows the Drill-Up button, and users can drill up.
CanDrill Down	If true, the control shows the Drill-Down button, and users can drill down.

SPEAK has a number of already configured DurationItems:

- There are Duration items for Hour, Day, Week, Month, Quarter, and Year.

- The Year item has CanDrillUp "False".

- The Hour item has CanDrillDown "False".

- All other settings of CanDrillUp and CanDrillDown are "True".

You specify Duration and Unit like this:

DurationItem	Duration	Unit
Hour	None	Hour
Day	Day	Hour
Week	Week	Day
Month	Month	Day
Quarter	Quarter	Week

Year	Year	Month

4.6.2   Drilling up and down

Drilling down means that the control changes the duration to the current unit. For example, if the current duration is Week and the current unit is Day, the control drills down by setting the duration to Day and refreshing the data it shows.

The control always shows a drill-down button, but it only enables this button in these conditions:

- The CanDrillDown of the current DurationItem is true

- The user has selected a segment

Drilling up means that the control changes the duration to the duration that is one magnitude higher. For example, if the current duration is Week, the control drills up by setting the duration to Month and refreshing the data it shows.

The control always shows a drill-up button, but it only enables it when there is a duration of a higher magnitude.

Note that the sequence of durations can be different when users drill up and when they drill down. This happens, for example, when you use the preconfigured DurationItems. If a user drills down while the duration is Month, the duration will change to Day. When the user drills up while the duration is Day, the duration will change to Week.

4.6.3   Details Panel

You can specify that SPEAK shows a Details Panel when users click on a segment in the JourneyChart control.

- Set the SegmentSelectedAction property of the JourneyChart control to ShowDetails

- Add a control, for example a ListControl, to the DetailsPanel placeholder

You have to populate the control you have put in the DetailsPanel placeholder yourself. You can do this in PageCode. The following statement checks that then user has selected a segment and that the SegmentSelectedAction is "ShowDetails":

```
if (selectedSegment && selectedSegment.action === "ShowDetails") {
 // Log to the console the data object for selected segment
console.log(selectedSegment.dataObject);
}
```

4.6.4   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
ShowAxis	If true, the control shows axes.	True
ShowLegend	If true, the control shows a legend for the	True

	chart.	
ShowValues	If true, the control shows the values of the data points inside the graph itself.	False
SingleChartHeight	Specify the height in pixels of the chart. If the chart is part of sequence of charts, the control gives each individual chart this height.	400
SingleChartWidth	If you specify "100%" the chart takes up the available width. If you specify a number, the control gives the chart this width in pixels. If the chart is part of sequence of charts, "100%" means each chart has the entire available width, and the chart sequence becomes a vertical sequence. If the chart is part of sequence of charts, and you specify an integer, the control gives each chart this width in pixels, and the sequence of charts wraps horizontally.	100%
StackSeries	If true, the control stacks the distinctive series in a multiseries chart.	False
Tooltip	Specify the text that you want displayed when the user hovers the mouse over the chart.	None
VisibleCategoriesRange	If a multi-series chart and this number is > 0, the control will show this number of categories along the horizontal axis. If the data contains more categories, users have to scroll horizontally. If this number is 0, the control shows all categories.	0
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource		
DurationItems	Specify a folder with DurationItems. The control has a Duration for each DurationItem in the folder.	
IsDurationButtonsPanelVisible	Is true, the control shows a panel with a button for each Duration. The control uses the Display Name of a DurationItem as a label for the button for that DurationItem.	
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
SegmentSelectedAction	Specify what happens when users click a segment in the chart: • NoAction: nothing happens • DrillDown: the control drills down	

	to the next duration level. See the description of drill-down for further information. • ShowDetails: the control shows a Details Panel for the segment.	
SelectedDurationItemId	Specify the DurationItem selected by the control when the page it is in opens.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.ChartDataProvider.Data}	
DataMapping		
CategoryChartField	Point to a ChartField item. The control uses this field as the X (horizontal) axis of the chart.	
CategoryFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values to filter the data column the CategoryChartField points to.	
SeriesChartField	Point to a ChartField. The control uses this field as the definition of the series it shows.	
SeriesFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values as a facet of the data	

	column the SeriesChartField points to. If you do not enter anything, the control will create a facet for each distinct value in the data column.	
ValueChartFields	Point to a ChartField item. The control uses this field as the Y (vertical) axis of the chart.	
Design		
Id	Lets you set the ID of the control.	

4.6.5 Configuration

You can create a configuration item for a JourneyChart by creating an item based on the JourneyFCChart Parameters template.

The configuration item has these specific fields:

- SegmentSelectedAction
- DurationItems
- SelectedDurationItemId
- IsDurationButtonsPanelVisible

It inherits these fields:

- ShowAxis
- ValueChartFields
- SingleChartWidth
- CategoryChartField
- ShowLegend
- Data
- ShowValues
- CategoryFilter
- SingleChartHeight
- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id
- VisibleCategoriesRange
- SeriesChartField
- SeriesFilter
- StackSeries

4.7 LineChart

The LineChart control shows a line chart. A line chart displays information as a series of data points connected by straight-line segments.

The LineChart control supports multi-axis charts. A multi-axis chart is a line chart that plots different values against the same category. Because each line in a multi-axis line chart is a different category, there is an axis for each value. The LineChart control shows a multi-axis line chart if you point it to a folder with multiple ChartField items in the ValueChartFields property.

You can specify in the ChartField items if the LineChart control shows the axis for the value on the left or on right side of the chart.

The LineChart control supports multiseries charts.

The AreaChart control supports showing a sequence of charts when the data it receives from the data provider contains multiple data sets.

4.7.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
ShowAxis	If true, the control shows axes.	True
ShowLegend	If true, the control shows a legend for the chart.	True
ShowValues	If true, the control shows the values of the data points inside the graph itself.	False
SingleChartHeight	Specify the height in pixels of the chart. If the chart is part of sequence of charts, the control gives each individual chart this height.	400
SingleChartWidth	If you specify "100%, the chart takes up the available width. If you specify a number, the control gives the chart this width in pixels. If the chart is part of sequence of charts, "100%" means each chart has the entire available width, and the chart sequence	100%

	becomes a vertical sequence. If the chart is part of sequence of charts, and you specify an integer, the control gives each chart this width in pixels, and the sequence of charts wraps horizontally.	
Tooltip	Specify the text that you want displayed when the user hovers the mouse over the chart.	None
VisibleCategoriesRange	It this number is > 0, the control will show this number of categories along the horizontal axis. If the data contains more categories, users have to scroll horizontally. If this number is 0, the control shows all categories.	0
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource		
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies	False

	based on the data source of the control.	
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.DataProvider.Data}	
DataMapping		
CategoryChartField	Point to a ChartField item. The control uses this field as the X (horizontal) axis of the chart.	
CategoryFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values to filter the data column the CategoryChartField points to.	
SeriesChartField	Point to a ChartField. The control uses this field as the definition of the series it shows.	
SeriesFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values as a facet of the data column the SeriesChartField points to. If you do not enter anything, the control will create a facet for each distinct value	

	in the data column.	
ValueChartFields	Point to a ChartField item. The control uses this field as the Y (vertical) axis of the chart. You can specify multiple ChartField items as a comma-separated list. The LineChart control will show a graph and an axis for each value.	
Design		
Id	Lets you set the ID of the control.	

4.7.2   Configuration

You can create a configuration item for a LineChart by creating an item based on the LineChart Parameters template.

The configuration item has these inherited fields:

- ShowAxis

- ValueChartFields

- SingleChartWidth

- CategoryChartField

- ShowLegend

- Data

- ShowValues

- CategoryFilter

- SingleChartHeight

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

- VisibleCategoriesRange

- SeriesChartField

- SeriesFilter

4.8 PieChart

The PieChart control shows a pie chart. A pie chart displays values as percentages of a whole, and each value is an individual slice of the whole. Pie charts are functionally identical to doughnut charts.

The PieChart control does not support multiseries charts.

The PieChart control supports showing a sequence of charts when the data it receives from the data provider contains multiple data sets.

4.8.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
ShowLegend	If true, the control shows a legend for the chart.	True
ShowValues	If true, the control shows the values of the data points inside the graph itself.	False
SingleChartHeight	Specify the height in pixels of the chart. If the chart is part of sequence of charts, the control gives each individual chart this height.	400
SingleChartWidth	If you specify "100%, the chart takes up the available width. If you specify a number, the control gives the chart this width in pixels. If the chart is part of sequence of charts, "100%" means each chart has the entire available width, and the chart sequence becomes a vertical sequence. If the chart is part of sequence of charts, and you specify an integer, the control gives each chart this width in pixels, and the sequence of charts wraps	100%

	horizontally.	
Tooltip	Specify the text that you want displayed when the user hovers the mouse over the button.	None
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource		
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies	False

	based on query string parameters passed in the URL.	
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.DataProvider.Data}	
DataMapping		
CategoryChartField	Point to a ChartField item. The control uses this field labels for the slices.	
CategoryFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values to filter the data column the CategoryChartField points to.	
ValueChartFields	Point to a ChartField item. The control uses this field as values of the slices.	
Design		
Id	Lets you set the ID of the control.	

4.8.2   Configuration

You can create a configuration item for a PieChart by creating an item based on the PieChart Parameters template.

The configuration item has these inherited fields:

- ShowAxis

- ValueChartFields

- SingleChartWidth

- CategoryChartField

- ShowLegend

- Data

- ShowValues

- CategoryFilter

- SingleChartHeight

- IsVisible

- AccessKey

- Tooltip

- Behaviors
- Id

4.9 RadarChart

The RadarChart control shows a radar chart. A radar chart consists of a sequence of equi-angular spokes, with each spoke representing a category. The data length of each spoke is proportional to the value of the data point for that category.

A radar chart is also known as a web chart, a spider chart, a star chart, a star plot, and other names.

The RadarChart control supports multiseries charts.

The RadarChart control supports showing a sequence of charts when the data it receives from the data provider contains multiple data sets.

4.9.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
ShowAxis	If true, the control shows axes.	True
ShowLegend	If true, the control shows a legend for the chart.	True
ShowValues	If true, the control shows the values of the data points inside the graph itself.	False
SingleChartHeight	Specify the height in pixels of the chart. If the chart is part of sequence of charts, the control gives each individual chart this height.	400
SingleChartWidth	If you specify "100%, the chart takes up the available width. If you specify a number, the control gives the chart this width in pixels. If the chart is part of sequence of charts, "100%" means each chart has the entire available width, and the chart sequence becomes a vertical sequence. If the chart is part of	100%

	sequence of charts, and you specify an integer, the control gives each chart this width in pixels, and the sequence of charts wraps horizontally.	
Tooltip	Specify the text that you want displayed when the user hovers the mouse over the chart.	None
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource		
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies	False

	based on rendering parameters passed to the presentation component.	
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.DataProvider.Data}	
DataMapping		
CategoryChartField	Point to a ChartField item. The control uses this field as spokes.	
CategoryFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values to filter the data column the CategoryChartField points to.	
SeriesChartField	Point to a ChartField. The control uses this field as the definition of the series it shows.	
SeriesFilter	Specify one or more values like this: "{value1,value2}". The control uses each of these values as a facet of the data column the SeriesChartField points to. If you do not enter anything, the control will create a facet for each distinct value in the data column.	
ValueChartFields	Point to a ChartField item. The control uses this field as the angles.	
Design		
Id	Lets you set the ID of the control.	

4.9.2   Configuration

You can create a configuration item for a RadarChart by creating an item based on the RadarChart Parameters template.

The configuration item has these inherited fields:

- ShowAxis

- ValueChartFields

- SingleChartWidth

- CategoryChartField

- ShowLegend

- Data

- ShowValues

- CategoryFilter

- SingleChartHeight

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

- VisibleCategoriesRange

- SeriesChartField

- SeriesFilter

4.10   Configuration Item Templates for Chart Controls

Many of the controls in the Chart group use configuration items based on common templates. This section describes these common configuration item templates.

4.10.1   The ChartField Template

All charting controls have properties called CategoryChartField, ValueChartFields and SeriesChartField in the DataMapping group. You specify a ChartField configuration item as the value of these properties, and you specify which chart data fields the control uses, as well as other information, in this ChartField configuration item.

You create a ChartField configuration item by creating an item based on the ChartField template. The convention is that place this item in the PageSettings folder of your page or application.

The item has these fields:

Field	Description
HeaderText	Specify the name the control shows as a label for this field.
DataField	Specify the name of the data field.
MultiAxisChartPosition	Specify the position of the axis the control shows for this field in a MultiAxis Line chart. You can specify Left, or Right.
CombinationChartRenderAs	Specify a combination chart rendering mode. You can specify Line, Area, orColumn.-
Prefix	Specify a string the control prepends to the chart values when it shows them.
Suffix	Specify a string the control appends to the chart values when it shows them.
NumberScale	Specify a NumberScale configuration item the control uses for scaling numbers. You can specify one of these predefined items: DataSize, Number, or Time. See the description of the NumberScale template for more details. You will also find information about how to create your own number scales.

4.10.2   The NumberScale Template

You use a NumberScale configuration item to scale numbers a charting control shows. Number scaling shortens numbers on charts according to a scale. You can specify a specific number scaling for each chart field in a ChartField configuration item.

A number scaling specification consists of two series of values: the ScaleValue and the ScaleUnit. The ScaleValue series is a series of numbers, and the ScaleUnit series if a series of labels. The two series must have the same number of members.

For example:

The predefined Number configuration item has ScaleValue "1000,1000,1000" and ScaleUnit "K,M,B". This is a standard Metric scaling: K for thousands (kilo), M for millions, and B for billions. This configuration will scale numbers for display like this:

Number	Scales as
800	800

Number	Scales as
8000	8K
8000000	8M
8000000000	8B

This means that values in the same chart can have different scales (if, for example, 8,000 and 8,000,000 were values in the same chart).

The logic is as follows:

1. If the value is less than the first number in ScaleValue, nothing happens. "800" scales as "800".

2. If the value is larger than the first number in ScaleValue, but less than the product of the first and the second number in ScaleValue, the value is divided by the first number in ScaleValue, and the first ScaleUnit is appended. 8000 is larger than 1000 but less than 1000*1000. Therefore, 8000 is divided by 1000 and K is appended: "8000" scales as "8K".

3. If the value is larger than the second number in ScaleValue, but less than the product of the first, second and third number in ScaleValue, the value is divided by the product of the first and the second number in ScaleValue, and the second ScaleUnit is appended. 8000000 is larger than 1000*1000 but less than 1000*1000*1000. Therefore, 8000000 is divided by 1000*1000 and M is appended: "8000000" scales as "8M".

4. And so forth.

SPEAK has three predefined NumberScale configuration items:

Name	ScaleValue	ScaleUnit
Number	"1000,1000,1000"	"K,M,B"
DataSize	"8,1024,1024,1024,1024"	"bytes,KB,MB,GB,TB"
Time (scales a value in seconds)	"60,60,24,7"	"min,hr,day,wk"

You can create your own scales for numbers this way:

1. Create a an item based on the NumberScale in the /sitecore/client/Business Component Library/System/Charting/NumberScales folder.

2. Enter a comma-separated and quoted string of numbers in the ScaleValue field.

3. Enter a comma-separated and quoted string text strings in the ScaleUnit field. It is important that you specify the exact same number of numbers in ScaleValue and text strings in ScaleUnit.

4. Save this item.

You can now select the NumberScale item you created in the NumberScale field of a ChartField item.

Chapter 5

Common

5.1 ActionControl

The ActionControl control shows a button that users click to open a panel with actions. Users trigger an action by clicking an action item, and you can specify what the action is (for example, a JavaScript that opens a new dialog). You use the ActionControl to give users quick access to a set of actions that are useful in the context they are in.

The panel is similar to a "ribbon," and you can organize the panel in columns and groups. Users can mark actions as favorites by clicking the "star" icon that the ActionControl always displays next to the action items. SPEAK displays favorites as buttons to the left of the button that opens the panel. Users can remove a favorite by clicking the star icon again. Favorites are stored for the current page in each user profile.

You cannot add actions to an ActionControl "on-the-fly" (from PageCode or other code) because ActionControls are created and rendered on the server.

5.1.1 How to create an ActionControl:

Follow these steps to use an ActionControl on a page:

1. Add an ActionControl component to the page. Usually you add the control to a container that you can set vertical alignment for, because the ActionControl works best when it is right-aligned.

2. Create a folder for configuration items under PageSettings.

3. Add an ActionControl item to this folder. This item is the start of the structure that forms the menu. You do not need to set any values in this item.

4. Add ActionColumn items as children of the ActionControl item. Add one ActionColumn item for each column you want in the menu. You do not need to set any values in this item.

5. Add an ActionGroup item as a child of an ActionColumn item for each group heading you need. Specify the heading text in the Text field of the item.

6. Add Action items as child items of the ActionGroups. Add one Action item for each action you want to add to the menu. Each item shows a menu item that users can click. Set the name that it shows in the Title field and the action in the Click field.

7. Bind the Data Source property of the ActionControl *component* to the ActionControl *item* you created in PageSettings.

5.1.2 How to enable and disable actions

The ActionControl enables all actions it contains by default. You can disable and enable individual actions like in several different ways:

8. You can disable an Action by selecting the Disable checkbox of the Action item, and enable it by deselecting the checkbox.

9. You can use the "actions" array in the View model of the ActionControl. For example:

```
this.ActionControl1.viewModel.actions()[0].disable()
```

to disable the first action in the array. Use "enable" to enable it.

10. You can use the getAction method of the View model. For example:

```
this. ActionControl1.viewModel.getAction('[data-sc-actionId=\"Sitecore
ID of Action item\"]\').disable()
```

to disable the Action where the Action item has the Sitecore ID in the call. Use "enable" to enable it.

5.1.3   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the ActionControl does not take up space on the page.	True
Menu		
Text	Specifies the text on the button.	
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button.	None.
	If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Specify the ActionControl item you use to define the content of the ActionControl panel.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true.	False
	Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on	False

	whether the user has authenticated.	
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
IsChecked	True if the CheckBox is checked.	True
Text	Use this to create a label for the CheckBox. The ActionControl shows the label to the right of the CheckBox, and you cannot change the position or the style. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
Value	You can use this property to associate a text string with the CheckBox. You can use this text as the "value" of the CheckBox. The CheckBox Parameters template includes the Value property, so you can use the technique described in the DataSource property to avoid enyering a static text string in this property.	"on"
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

5.1.4   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
text	
isOpen	
actions	
favorites	
userProfileKey	
actionsStatus	

5.1.5 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
focus	
getAction	
hide	
invokeAction	
invokeFavorite	
listenTo	
listenToOnce	
once	
set	
setFavoritesInUserProfile	
show	
stopListening	
toggle	
toggleFavorite	
toggleIsOpen	
updateFavorites	
updateActionsStatus	

5.1.6 Configuration

You can create a configuration item for an ActionControl by creating an item based on the ActionControl Parameters template.

The configuration item has these specific fields:

Text - this field corresponds to the Text property and you use the field to make it easier to translate UI texts.

Menu

It inherits these fields:

- IsVisible
- AccessKey
- Tooltip
- Behaviors
- I

5.2 Button

A Button control presents a standard button that users can click. If you set the ButtonType property to "primary," SPEAK makes it stand out visually. You can set the size of a button to "normal" or "large" with the Dimensions property.

You should set Buttons that users click to carry out the main purpose of a page as "primary" and use "default" for other buttons. You can also set the ButtonType to "inverse." An "inverse" button uses the same colors as a "default" button, but swaps the text and background colors.

Buttons can display icons as well as text, and they can show both at the same time.

You specify icons using the ImageUrl property. You can use the same icons here as when you select an icon for an item in the content tree. Button controls also support defining the icon as a "sprite," selected from a sprite sheet. The ImageUrl property stores the URL of the sprite sheet, and the BackgroundPosition property stores the coordinates of the desired sprite (icon/image).

If you use both icon and text, SPEAK positions the icon to the left of the text. You cannot change this position.

5.2.1 Properties

Name	Description	Default
Appearance		
BackgroundPosition	Lets you set the offset of the sprite/icon/image in a sprite sheet file in this format: -<x>px -<y>px Where <x> indicates the horizontal number of pixels from the left edge of the sprite sheet to the left edge of the sprite (represented as a negative number), and <y> indicates the vertical number of pixels from the top edge of the sprite sheet (represented as a negative number) to the top edge of the sprite.	-0px -0px
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
ButtonType	You can choose between Default, Primary and Inverse.	Default
Dimension	Lets you set the size of the button. You can choose between Normal and Large.	Normal
ImageUrl	Lets you set the Sitecore path of an icon (for example, "Network/16x16/home.png" for the icon shown for the "/sitecore/content/home" item) or sprite sheet. SPEAK provides a number of sprite sheets in the "\sitecore\shell\client\Speak\Assets\img" folder (found in the Website folder).	
IsEnabled	If false, the Button does not respond to clicks.	True

IsVisible	If false, the Button does not take up space on the page.	True
Text	Lets you set the text you want displayed on the Button. You can enter a static text, set the text from PageCode, or you can bind to a property of another control. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the Button Parameters template, and point to this item here. You can specify the text and tooltip that the button displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False

VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	
Events		
Click	Lets you specify JavaScript that SPEAK executes when a user clicks the button.	

5.2.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
isEnabled	
text	
backgroundPosition	
isOpen	

5.2.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
click	
disable	
enable	
focus	
hide	
listenTo	
listenToOnce	
once	

Name	Description
set	
show	
stopListening	
toggle	

5.2.4 Configuration

You can create a configuration item for a Button by creating an item based on the Button Parameters template.

The configuration item has these specific fields:

- ButtonType

- Dimension

- Text – this field corresponds to the Text property, and you use the field to make it easier to translate UI texts.

- ImageUrl

- BackgroundPosition

It inherits these fields:

- Click

- IsEnabled

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

5.3 CheckBox

A CheckBox control can be selected ("checked") or not.

You use one or more CheckBoxes to let users select one or more options. You should use radio buttons when the options are mutually exclusive, and you want users to select only one option at a time.

5.3.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsEnabled	If false, the CheckBox does not respond to clicks.	True
IsVisible	If false, the CheckBox does not take up space on the page.	True
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the CheckBox Parameters template, and point to this item here. You can specify the text and tooltip that the CheckBox displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the	False

	generated HTML. For more information on caching, refer to the SDN.	
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
IsChecked	True if the CheckBox is checked.	True
Text	Use this to create a label for the CheckBox. The label is shown to the right of the CheckBox, and you cannot change the position or the style. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
Value	You can use this property to associate a text string with the CheckBox. You can use this text as the "value" of the CheckBox. The CheckBox Parameters template includes the Value property, so you can use the technique described in the DataSource property to avoid entering a static text string in this property.	"on"
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

5.3.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	

Name	Additional Description
isEnabled	
text	
value	
isChecked	

5.3.3   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
disable	
enable	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
show	
stopListening	
toggle	

5.3.4   Configuration

You can create a configuration item for a CheckBox by creating an item based on the CheckBox Parameters template.

The configuration item has these specific fields:

- Value
- IsChecked
- Text - this field corresponds to the Text property, and you use the field to make it easier to translate UI texts.

It inherits these fields:

- IsEnabled
- IsVisible
- AccessKey

- Tooltip
- Behaviors
- Id

5.4 ChildRenderer

The ChildRenderer control renders all items that are children of the item you specify in the DataSource property. It is a pure server-side control, and you cannot use data binding or JavaScript.

It renders items "as is," so the items need to have layouts. If you use SPEAK components to render the items, these SPEAK components are rendered just as other SPEAK components are, and they work the same way.

5.4.1 Properties

Name	Description	Default
Appearance		
After	Specify an HTML fragment the control inserts after each item.	</div>
Before	Specify an HTML fragment the control inserts before each item.	<div>
Behavior		
DataSource	You can specify a Sitecore item, and all child items of this item are rendered by the ChildRenderer. You cannot filter the child items.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
UsePlaceholders	If True, the control uses placeholders when it is rendering. If False, it renders items as a list.	False
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False

| VaryByQueryString | Lets you specify whether output caching varies based on query string parameters passed in the URL. | False |
| VaryByUser | Lets you specify whether output caching varies by the domain and username of the context user. | False |

5.5 ComboBox

The ComboBox control shows a drop-down list of items. When a user selects an item from the list, the ComboBox control returns the value of the selected item (as "selectedValue").

You bind to a data source in the Items property, and the ComboBox control shows all the items that the data source retrieves.

The ComboBox shows the display names of the items by default, but you can use the DisplayFieldName property of the ComboBox to select another item field to show.

SPEAK selects the first (top) item in the ComboBox control by default. This means that the *selectedItem* member never returns null if the ComboBox has any items at all.

You can set properties of a ComboBox control programmatically in different ways. We recommend that use the *rebind* function. It has this syntax:

```
rebind (items, selectedItem, selectedValue, displayFieldName, valueFieldName)
```

If you provide both *selectedValue* and *selectedItem*, the ComboBox control will use *selectedItem*.

5.5.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsEnabled	If false, users cannot select items.	True
IsVisible	If false, the ComboBox does not take up space on the page.	True
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
WidthPolicy	Specify how SPEAK interprets the width of the control. You can specify: • Default: the default width of the control is used. • BlockLevel: the width is set relative to the width of the container.	Default.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	

DataSource	You can specify a Sitecore item, and all child items of this item are shown by the ComboBox. You cannot filter the child items.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
DisplayFieldName	Specify the name of the item field the ComboBox control shows.	$displayName
Items	You can bind to a data source and the ComboBox shows the items retrieved by this data source.	itemId
SelectedItem	Specify the selected item by name.	The first (top) item
SelectedValue	Specify the selected item by value (ID).	
ValueFieldName	Specify the data field the ComboBox uses as the option value (and returns as *selectedValue*).	ID of the item.
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example, SelectDepartment).	

5.5.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
isEnabled	
Items	
selectedItem	
selectedValue	
valueFieldName	
displayFieldName	
selectedOptions	
selectedItems	Deprecated in favor of selectedItem
selectedItemId	Deprecated in favor of selectedValue

5.5.3   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
disable	
enable	
focus	
getDisplayFieldName	
getItemFromValue	
getValueFieldName	
hide	
isPopulatedArray	
listenTo	
listenToOnce	
once	
rebind	
selectItemsChangedHandler	

Name	Description
set	
setSelection	
show	
stopListening	
toggle	

5.5.4   Configuration

You can create a configuration item for a ComboBox by creating an item based on the ComboBox Parameters template.

The configuration item has these specific fields:

- DisplayFieldName

It inherits these fields:

- IsEnabled
- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id

5.6 DatePicker

The DatePicker lets users pick a date. It displays a text box with the currently selected date (if any), and when users click inside the text box, a "date picker" opens. When the user clicks on a date in the date picker, this date becomes the selected date. Users can also type a date directly into the text box, and the DatePicker goes to this date.

You can set a date that is the default "selected date" by setting the Date property.

5.6.1 Setting and getting the date from JavaScript

You can set and get date in different way in JavaScript (in PageCode, for example).

Setting the date

You can set date in two different ways:

`Model.set("date")`

This sets the date in Sitecore DateFormat.

Code example:

```
this.DatePicker1.set("date", "20131224")
```

`Model.set("formattedDate")`

This sets the date in the specified format.

Code example:

```
this.DatePicker1.set("formattedDate", "2013/12/24")
```

Note

A third function, `Model.viewModel.setDate()`, corresponds to `set("formattedDate")` and we recommend that you use `Model.set`.

Getting the date

You can get the date in three different ways:

`Model.get("date")`

This returns the date in Sitecore DateFormat, for example: `20131224`.

Code example:

```
this.DatePicker1.get("date")
```

`Model.get("formattedDate")`

This returns the date in the format specified, for example: `2013/12/24`.

Code example:

```
this.DatePicker1.get("formattedDate")
```

`Model.viewModel.getDate()`

This returns the date as a JavaScript DateObject, for example: `Tue Dec 24 2013 00:00:00 GMT+0100 (Romance Standard Time)`.

Code example:

```
this.DatePicker1.viewModel.getDate()
```

5.6.2   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsEnabled	If true, the control is enabled.	True
IsVisible	If true, the control is visible.	True
ToolTip	The control displays this text when users hovers the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on	False

	rendering parameters passed to the presentation component.	
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Date	You use this property to set a date, and you also use it to retrieve the currently selected date. If you set a date, this date is the selected date initially. The DatePicker controls only accepts dates in Sitecore date format, such as: "20131224", "20131224T123456" and "20131224T123456:635280071319116097"	The current date.
FirstDay	You select a day of the week, from Monday to Sunday. The DatePicker uses the day you select as the first day of the week.	Monday
Format	The date format. Specify the format using these specifiers: d: day of month (no leading zero) dd: day of month (two digits) o: day of the year (no leading zeros) oo: day of the year (three digits) D: day name short DD: day name long m: month of year (no leading zero) mm: month of year (two digits) M: month name short MM: month name long yy: year (two digits) yyyy: year (four digits) @: Unix timestamp (milliseconds since 01/01/1970) !: Windows ticks (100-nanosecond periods since 01/01/0001)	dd/mm/yyyy
IsReadOnly	If True, users can only select dates from the DatePicker calendar dropdown, and they cannot enter dates directly by typing.	False
ShowOtherMonths	If True, the DatePicker shows 35 days, and some of these are in the previous or in the next month. If False, only the current month is shown in the DatePicker.	True
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example, CloseButton).	

5.6.3 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the

"IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
date	
dateFormat	
firstDay	
showOtherMonths	
dayNamesMin	
isReadOnly	
isEnabled	

5.6.4 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
dialog	
focus	
getDate	
hide	
initialized	
isDisabled	
listenTo	
listenToOnce	
once	
raiseEvent	
refresh	
set	
setDate	
show	
stopListening	
toggle	
widget	

5.6.5   Configuration

You can create a configuration item for a DatePicker by creating an item based on the DatePicker Parameters template.

The configuration item has these specific fields:

- Date

- Format

- FirstDay

- ShowOtherMonths

- IsReadOnly

It inherits these fields:

- IsEnabled

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

5.7 DropDownButton

A DropDownButton control presents a button that users can click. It has two states: open and closed. The DropDownButton shows the current state with an arrow. When it is in the open state, the arrow points up, when it is in the closed state, the arrow points down. The DropDownButton toggles between these two states each time a user clicks the button.

A DropDownButton control adds a placeholder called Content. It shows all controls that you put in this placeholder when it is in the open state. You can add any SPEAK component to this placeholder. Note that when users click any control in this placeholder, the "click" event is raised for both the clicked control and the DropDownButton. This means that the DropDownButton toggles to the closed state.

If you set the ButtonType property to Primary, SPEAK makes it stand out visually. You can set the size of a button to normal or large with the Dimensions property.

You should set buttons that users click to carry out the main purpose of a page as primary, and you should use default for other buttons. You can also set the ButtonType to inverse. An inverse button uses the same colors as a default button, but swaps the text and background colors.

Buttons can display icons as well as text and they can show both at the same time.

You specify icons using the ImageUrl property. You can use the same icons here as when you select an icon for an item in the content tree. Button controls also support defining the icon as a "sprite," selected from a sprite sheet. The ImageUrl property stores the URL of the sprite sheet, and the BackgroundPosition property stores the coordinates of the desired sprite (icon/image).

If you use both icon and text, SPEAK positions the icon to the left of the text. You cannot change this position.

5.7.1 Properties

Name	Description	Default
Appearance		
BackgroundPosition	Lets you set the offset of the sprite/icon/image in a sprite sheet file in this format: -\<x>px -\<y>px Where \<x> indicates the horizontal number of pixels from the left edge of the sprite sheet to the left edge of the sprite (represented as a negative number), and \<y> indicates the vertical number of pixels from the top edge of the sprite sheet (represented as a negative number) to the top edge of the sprite.	-0px -0px
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
ButtonType	You can choose between Default, Primary and Inverse.	Default
ContentAlignment	Specify how the content in the Content is aligned: Left or Right.	Left
ImageUrl	Lets you set the Sitecore path of an icon (for example,	

	"Network/16x16/home.png" for the icon shown for the "/sitecore/content/home" item) or sprite sheet. SPEAK provides a number of sprite sheets in the "\sitecore\shell\client\Speak\Assets\img" folder (found in the Website folder).	
IsEnabled	If false, the Button does not respond to clicks.	True
IsVisible	If false, the Button does not take up space on the page.	True
Text	Lets you set the text you want displayed on the Button. You can enter a static text, set the text from PageCode, or you can bind to a property of another control. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the DropDownButton Parameters template, and point to this item here. You can specify the text and tooltip that the button displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False

VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example, CloseButton).	
Events		
Click	Lets you specify JavaScript that SPEAK executes when a user clicks the button.	

5.7.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
isEnabled	
text	
imageUrl	
backgroundPosition	
isOpen	

5.7.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
_changeStatus	
addBehavior	
click	

Name	Description
close	
collapse	
disable	
enable	
expand	
focus	
hide	
listenTo	
listenToOnce	
once	
open	
set	
show	
stopListening	
toggle	
toggleEnable	
toogle	

5.7.4   Configuration

You can create a configuration item for a DropDownButton by creating an item based on the DropDownButton Parameters template.

The configuration item has these specific fields:

- ButtonType
- Text – this field corresponds to the Text property, and you use the field to make it easier to translate UI texts.
- ImageUrl
- BackgroundPosition
- ContentAlignment

It inherits these fields:

- Click
- IsEnabled
- IsVisible
- AccessKey
- Tooltip
- Behaviors

- Id

5.8 GlobalLogo

You use the GlobalLogo control to show an image that users can click, and you can set a destination URL that the browser navigates to when users click the control.

You use the GlobalLogo to provide a consistent way for users to navigate to a specific page from all pages in a SPEAK application. The StartButton placeholder of the GlobalHeader substructure control is the recommended place for a GlobalLogo.

5.8.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
Dimension	Lets you set the size of the button. You can use these sizes: id16x16 id24x24 id32x32 id48x48	id24x24
ImageUrl	Lets you set the Sitecore path of an icon (for example, "Network/16x16/home.png" for the icon shown for the "/sitecore/content/home" item) or sprite sheet. SPEAK provides a number of sprite sheets in the "\sitecore\shell\client\Speak\Assets\img" folder (found in the Website folder).	
IsVisible	If false, the Button does not take up space on the page.	True
NavigateUrl	Specifies a resource that the page should navigate to when the button is clicked. You can specify a Sitecore path to an item, for example: "/sitecore/client/sitecore/applications/sample/pages/home" to navigate to another page in your application. You can also specify a URL to a web resource, for example: "http://www.sitecore.net".	
Source	The Sitecore path of the image.	
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
Behavior		

AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Not relevant for this control.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

5.9 HyperlinkButton

You can use a HyperlinkButton control in two different ways:

- You can specify the URL of a resource that users go to when they click the button. You use the NavigateUrl property for this.

- You can specify a JavaScript that SPEAK executes when users click the button. You use the Click property for this.

The NavigateUrl property takes precedence over Click. If the NavigateUrl property has a value, the HyperlinkButton does execute the JavaScript specified in the Click property.

A HyperlinkButton control looks like a hyperlink on a webpage. You can set the text that it displays.

HyperlinkButtons can be part of a HyperlinkButtonsGroup container.

5.9.1 Properties

Name	Description	Default
Appearance		
Behaviors	The component does not use this property.	
IsEnabled	If false, the HyperlinkButton does not respond to clicks.	
IsVisible	If false, the HyperlinkButton does not take up space on the page.	
Text	Text displayed by the HyperlinkButton. You can enter a static text, set the text from PageCode, or you can bind to a property of another control. You should not specify a static text in the component properties if your dialog or application supports multiple languages. See the DataSource property for an alternative way to specify text.	
ToolTip	Text displayed when the user hovers the mouse over the HyperlinkButton. You should not specify a static text in the component properties if your dialog or application supports multiple languages. See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute, a single character.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the HyperlinkButton Parameters template, and point to this item here. You can specify the text and tooltip that the button displays in the	

	configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
NavigateUrl	Specifies a resource that the page should navigate to when the button is clicked. You can specify a Sitecore path to an item, for example: "/sitecore/client/sitecore/applications/sample/pages/about" to navigate to another page in your application. You can also specify a URL to a web resource, for example: "http://www.sitecore.net".	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Text Id of the placeholder where Sitecore renders this control.	
Target	Specifies a text string that the button renders as an HTML "target" attribute. SPEAK does not validate the string: it renders it as is. This property is ignored when NavigateUrl is not set, and it does not have any effect for JavaScript in the Click property.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN.	False
VaryByData	Controls whether output caching varies based on the data source of the control.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

Events		
Click	Specifies the JavaScript that is executed when a user clicks the button.	

5.9.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
isEnabled	
text	
isButtonMode	

5.9.3   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
click	
disable	
enable	
enabledChange	
focus	
hide	
listenTo	
listenToOnce	
once	
preventIfDisable	
set	
show	
stopListening	
toggle	

5.9.4 Configuration

You can create a configuration item for a HyperlinkButton by creating an item based on the HyperlinkButton Parameters template.

The configuration item has these specific fields:

- IsButtonMode
- NavigateUrl
- Target
- Text - this field corresponds to the Text property, and you use the field to make it easier to translate UI texts.

It inherits these fields:

- Click
- IsEnabled
- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id

5.10 IconButton

An IconButton has the same functionality as a regular button, but it displays only an icon (image) and never a text label. You specify an image to use as an icon in the ImageUrl property. You can use the same icons here as when you select an icon for an item in the content tree.

IconButton controls let you define the icon as a "sprite" selected from a sprite sheet. The ImageUrl property stores the URL of the sprite sheet, and the BackgroundPosition property stores the coordinates of the desired sprite (icon/image).

5.10.1 Properties

Name	Description	Default
Appearance		
BackgroundPosition	Indicates the offset of the sprite/icon/image in a sprite sheet file. The value must have the following format: -\<x>px -\<y>px Where \<x> indicates the horizontal number of pixels from the left edge of the sprite sheet to the left edge of the sprite (represented as a negative number), and \<y> indicates the vertical number of pixels from the top edge of the sprite sheet (represented as a negative number) to the top edge of the sprite.	-0px -0px
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
Dimensions	Lets you set the dimensions (in pixels) of the. Select between any of the following dimensions:: id16x16 id24x24 id32x32 id48x48 The Dimensions field always indicates the space taken up by the IconButton on the screen. If you specify Dimensions but do not specify the BackgroundPosition, the IconButton control resizes the image of the icon to fit the selected Dimensions. If you provide a BackgroundPosition, or if you do not specify Dimensions or BackgroundPosition, the IconButton crops the image to the dimensions of the button (which by default is 16x16).	id16x16
ImageUrl	The Sitecore path of the icon (for example, "Network/16x16/home.png" for the icon shown for the "/sitecore/content/home" item) or sprite sheet. SPEAK provides a number of sprite sheets in the	

	"\sitecore\shell\client\Speak\Assets\img" folder (found in the Website folder).	
isEnabled	If false, the IconButton is disabled. Disabled buttons are dimmed and do not respond to clicking.	True
isVisible	If false, the IconButton is invisible and it does not take up space in the layout. This means that if you change an IconButton control's visibility, other controls can change position on the screen.	True
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the IconButton Parameters template, and point to this item here. You can specify the tooltip that the IconButton displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	That is the standard Sitecore rendering parameters string. All properties in a SPEAK rendering are stored in the parameters string.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on	False

	rendering parameters passed to the presentation component.	
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseIconButton).	
Events		
Click	Activity for SPEAK to perform when the user clicks the IconButton.	

5.10.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
isEnabled	
imageUrl	
backgroundPosition	
Title	Corresponds to the ToolTip property.

5.10.3 View Members

View members represent events that the control either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
click	
disable	
enable	
focus	
hide	
listenTo	
listenToOnce	

Name	Description
once	
set	
show	
stopListening	
toggle	

5.10.4 Configuration

You can create a configuration item for a HyperlinkButton by creating an item based on the IconButton Parameters template.

The configuration item has these specific fields:

- ImageUrl
- Dimensions
- BackgroundPosition

It inherits these fields:

- Click
- IsEnabled
- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id

5.11   IconHyperlinkButton

An IconHyperlinkButton control is a hyperlink with an icon. SPEAK always shows the icon to the left of the text, and users can click both the icon and the text.

You can use an IconHyperlinkButton control in two different ways:

- You can specify the URL of a resource, and users can navigate to this resource by clicking the button. You use the NavigateUrl property for this.

- You can specify the JavaScript code that SPEAK executes when a user clicks the button. You use the Click property for this.

NavigateUrl takes precedence over Click. If the NavigateUrl property has a value, the IconHyperlinkButton does not execute the JavaScript specified in the Click property.

You specify an icon to use in the ImageUrl property. You can use the same icons here as, for example, when you select an icon for an item in the content tree.

IconHyperlinkButton controls support defining the icon as a "sprite" selected from a sprite sheet. The ImageUrl property stores the URL of the sprite sheet, and the BackgroundPosition property stores the coordinates of the desired sprite (icon/image).

5.11.1   Properties

Name	Description	Default
Appearance		
BackgroundPosition	Indicates the offset of the sprite/icon/image in a sprite sheet file. The value must have the following format: -<x>px -<y>px Where <x> indicates the horizontal number of pixels from the left edge of the sprite sheet to the left edge of the sprite (represented as a negative number), and <y> indicates the vertical number of pixels from the top edge of the sprite sheet (represented as a negative number) to the top edge of the sprite.	-0px -0px
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
ImageUrl	The Sitecore path of the icon (for example, "Network/16x16/home.png" for the icon shown for the "/sitecore/content/home" item) or sprite sheet. SPEAK provides a number of sprite sheets in the "\sitecore\shell\client\Speak\Assets\img" folder (found in the Website folder).	
IsEnabled	If false, the IconHyperlinkButton does not respond to clicks.	
IsVisible	If false, the IconHyperlinkButton does not take up space on the page.	

Text	Lets you set the text you want displayed on the IconHyperlinkButton. You can enter a static text, set the text from PageCode, or you can bind to a property of another control.	
	If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the IconHyperlinkButton.	None.
	If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
	Use a configuration item based on the IconHyperlinkButton Parameters template, and point to this item here. You can specify the text and tooltip that the button displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
NavigateUrl	Specifies a resource that the page should navigate to when the button is clicked.	
	You can specify a Sitecore path to an item, for example: "/sitecore/client/sitecore/applications/sample/pages/about" to navigate to another page in your application.	
	You can also specify a URL to a web resource, for example: "http://www.sitecore.net".	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Target	Specifies a text string that the button renders as an HTML "target" attribute. SPEAK does not validate the string: it renders it as is. This property is ignored when NavigateUrl is not set, and it does not have any effect for JavaScript in the Click property.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true.	False
	Sitecore generates SPEAK component HTML on the server	

	side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: NextStepButton).	
Events		
Click	Specifies the JavaScript that is executed when a user clicks the button.	

5.11.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
isEnabled	
text	
imageUrl	
backgroundPosition	

5.11.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description

Name	Description
click	
disable	
enable	
enabledChange	
focus	
hide	
preventIfDisable	
set	
show	
toggle	

5.11.4   Configuration

You can create a configuration item for an IconHyperlinkButton by creating an item based on the IconHyperlinkButton Parameters template.

The configuration item has these specific fields:

- Text
- ImageUrl
- BackgroundPosition
- NavigateUrl
- Target

It inherits these fields:

- Click
- IsEnabled
- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id

5.12 Image

You use the Image control to show an image on a SPEAK page.

5.12.1 Properties

Name	Description	Default
Appearance		
Alt	Specify a text that the Image control renders as an HTML "alt" attribute for the image.	
	If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
Height	Lets you set the height of the image in pixels.	
IsVisible	If false, the Image does not take up space on the page.	True
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button.	None.
	If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
Width	Lets you set the width of the image in pixels.	
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
	Use a configuration item based on the Image Parameters template, and point to this item here. You can specify the text and tooltip that the button displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	

Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
ImageUrl	Specify the Sitecore path of image.	
Design		
Id	Lets you set the ID of the control.	

5.12.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
imageUrl	
alt	
isVisible	
width	
height	
type	
name	

Name	Additional Description
src	

5.12.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
show	
stopListening	
toggle	

5.12.4 Configuration

You can create a configuration item for an Image control by creating an item based on the Image Parameters template.

The configuration item has these specific fields:

- Alt
- Height
- Width
- ImageUrl

It inherits these fields:

- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id

5.13   ItemRenderer

The ItemRenderer control renders a single Sitecore item. The item must have a layout.

The ItemRenderer control is a server-side control, so you cannot data bind any of the properties of the control, and you cannot reference the control in JavaScript in the client. If you have added SPEAK components to the layout, however, you have access to these components in the exact same way as if you had added them directly to the page.

5.13.1   Properties

Name	Description	Default
Behavior		
DataSource	Specify the item that the ItemRenderer renders. The item must have a layout.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
UsePlaceholders	Determines if the rendering uses placeholders or renders the items as a list.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False

5.13.2 Configuration

You can create a configuration item for an ItemRenderer by creating an item based on the ItemRenderer Parameters template.

The configuration item has this field:

- UsePlaceholders

5.14   Label

A Label control renders a caption for a SPEAK input control (for example, a TextBox or CheckBox). You associate the Label control with another SPEAK control by setting the Target property of the Label control.

You use the dedicated Label control instead of a Text control for two reasons:

- SPEAK can style and position the Label control differently

- Some browsers use the association between the Label control and the other control for enhancing accessibility

5.14.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
Target	Specify the ID of the control that the Label is associated with.	
Text	Specify the text that the control shows.	
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Type	Specify either "text" or "password".	text
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the Label Parameters template, and point to this item here.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the	False

	generated HTML. For more information on caching, refer to the SDN.	
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

5.14.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
text	

5.14.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
focus	
hide	
listenTo	
listenToOnce	
once	

Name	Description
set	
show	
stopListening	
toggle	

5.14.4   Configuration

You can create a configuration item for a Label control by creating an item based on the Label Parameters template.

The configuration item has these specific fields:

- Target
- Text

It inherits these fields:

- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id

5.15   ListBox

A ListBox control shows a list of items. You bind to a SearchDataSource, and the ListBox shows all the items that the search retrieves.

You can specify that the ListBox control uses the name of a different field of the items than the default display name, but you must keep the following in mind: If an item does not contain the field that you specify, there is no "fall-back" mechanism, and the ListBox control shows such an item as an empty string.

Users can select items in the list, and you can specify that users can make multiple selections. The ListBox returns the values of the selected items in the SelectedValue or SelectedValues member. If you have enabled multiple selections, SelectedValues contains a comma-separated list of values, while SelectedValue contains the value of the first (top) item the user has selected.

You can set properties of a ListBox control programmatically in different ways. We recommend that use the *rebind* function. It has this syntax:

```
rebind (items, selectedItem, selectedValue, displayFieldName, valueFieldName,
selectedItems, selectedValues)
```

If you provide two or more out of *selectedItem*, *selectedValue*, *selectedItems* and *selectedValues*, the order or precedence is: *selectedItems* > *selectedValues* > *selectedItem* > *selectedValue*.

5.15.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsEnabled	If false, the Button does not respond to clicks.	True
IsVisible	If false, the Button does not take up space on the page.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None
WidthPolicy	Specify how SPEAK interprets the width of the control. You can specify: Default: the default width of the control is used. BlockLevel: the width is set relative to the width of the container. The control fills the container horizontally.	Default
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Specify the ID of the item that is the root of the items you want the ListBox to show. The control shows all children of	

	this item. You can also bind to a data source such as the SearchDataSource.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
DisplayFieldName	Specify the name of a field to display instead of the default.	
Items	Bind to the items the control shows.	
Multiple	If true, users can select more than one item in the list at the same time.	False
SelectedItem	Get or set the selected item.	None
SelectedItems	Get or set the selected items as a comma-separated list.	None
SelectedValue	Get or set the selected item by value (ID).	None
SelectedValues	Get or set the selected item by value (ID) as a comma-separated list.	None
ValueFieldName	Specify the data field the ListBos uses as the option value (and returns as *selectedValue(s)*).	ID of the item.
Design		
Id	Lets you set the ID of the control.	

5.15.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
isEnabled	
items	
selectedItem	
selectedValue	
valueFieldName	
displayFieldName	
selectedOptions	
selectedItems	A list of all the currently selected items.
selectedItemId	The ID of the currently selected item.
multiple	
selectedValues	
selectedItemIds	A comma-separated list of IDs of all currently selected items.

5.15.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
_selectedItems	
addBehavior	
disable	
enable	
focus	
hide	
listenTo	
listenToOnce	
once	
set	

Name	Description
show	
stopListening	
toggle	

5.15.4 Configuration

You can create a configuration item for a ListBox control by creating an item based on the ListBox Parameters template.

The configuration item has these specific fields:

- Database

It inherits these fields:

- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id

5.16   MessageBar

The MessageBar control displays messages to the user. You can add links to actions to a message. You cannot change the styling of a MessageBar, and you cannot set the size of the control.

The MessageBar shows Message items that you specify in the Messages property. You can add multiple messages to one MessageBar control by adding a pipe-separated list of Message item IDs in the Messages property. In this case, the MessageBar shows the first message in the list of messages, and the rest of the messages are available when the user clicks a drop-down icon that the MessageBar automatically adds.

A Message item is a Sitecore item that is based on a MessageType template. There are three different MessageType templates:

- Error Message

- Warning Message

- Notification Message

The three types of messages are styled differently, but have the same basic functionality. They have these fields:

- Text – The message text

- IsClosable -- if True, users can close the message

- IsTemporary – if True, the message closes automatically after some time (only the Notification message type has this field)

If you add multiple messages to one MessageBar, the messages do not have to be of the same type.

A message can have one or more "action links." You add these links to messages like this:

1. Create an item based on the ActionsLink template

2. The item now has two fields:

    1) Text: the name of the link (this is what the user sees).

    2) Action: the action that is triggered when the clicks the link. For example, this can be JavaScript.

3. Add the item to the Action field of the Message.

You can add multiple action links to a message.

5.16.1   How to use the MessageBar

You should show the MessageBar when there is message. In other words, the MessageBar is a dynamic control that is only shown in certain circumstances.

You sometimes have to determine when to show the MessageBar in either PageCode or by using rules, but there are simple scenarios where you use other methods.

For example, you have a dialog that shows a search result, and you use a SearchDataSource to retrieve items. If the search does not find any items, you want to show a notification to the user. You add a MessageBar to the dialog and set IsVisible to False. You data bind the IsVisible property to the hasNoItems member of the SearchDataSource. When there are no items in the search result, the MessageBar becomes visible and shows the notification.

5.17 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the MessageBar does not take up space on the page.	True
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the MessageBar Parameters template, and point to this item here. You can specify the text and tooltip that the CheckBox displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False

VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Messages	Specify item IDs of the Message item the MessageBar control shows. You can specify multiple messages as a pipe-separated lost.	
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

5.17.1 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
errors	
warnings	
notifications	
expanded	
fadeVisible	
type	
name	
isVisible	
topMessageClass	
headText	
totalMessageCount	

5.17.2   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
show	
stopListening	
toggle	

5.17.3   Configuration

You can create a configuration item for a MessageBar by creating an item based on the MessageBar Parameters template.

The configuration item has these specific fields:

- Messages

It inherits these fields:

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

5.18   PageCode

The PageCode component is a basic component that you must add to all pages, and it must always use the Page.Code placeholder. It provides links to the CSS files and JavaScript files that all SPEAK pages use. If you do not add a PageCode component a page, the page does not look right – and it does not work. You must add one and only one PageCode component per page.

You also use the PageCode to point to folders with JavaScript files for the page, and to point to the type name and the assembly of server-side code for the page.

5.18.1   Script Files and Style Sheet Files

You specify a page script file in the PageCodeScriptFileName property. You can specify additional script files, and you can specify your own style sheets.

To add a script file:

1. Create a script file.

2. Add a configuration item based on the Page-Script-File item template to your PageSettings.

3. Specify the path to the script file in the File field of the configuration item. It must be a fully qualified path, for example: `/sitecore/shell/client/MyApp/MyScript.js`.

If you add more than one configuration item (because you need to add more than one script file), the script files will be included in the same order as the items have in PageSettings, and after a script file specified in the PageCodeScriptFileName property.

You can add multiple script files like this:

1. Create a folder and put your script files here.

2. Add a configuration item based on the Page-Script-Folder item template to your PageSettings.

3. Specify the path to the folder in the Folder field of the configuration item. It must be a fully qualified path, for example: `/sitecore/shell/client/MyApp/MyScripts`.

The files in the folder will be included in alphabetical order, based on their names. All files will be included a script file specified in the PageCodeScriptFileName property.

You can add cascading style sheets to you page. We do not recommend that you use this to modify the design of SPEAK pages and applications, but it can necessary to modify the styling of some controls in specific circumstances.

An example is the TextArea control. You can set Columns and Rows in the properties, but you often need to set the "width" style in pixels for it to look right.

To add a style sheet:

1. Create a cascading style sheet.

2. Add a configuration item based on the Page-Stylesheet-File item template to your PageSettings.

3. Specify the path to the style sheet file in the Stylesheet field of the configuration item. It must be a fully qualified path, for example: `/sitecore/shell/client/MyApp/MyStyle.css`.

If you add more than one configuration item (because you need to add more than one style sheet), the style sheets will be included in the same order as the items have in PageSettings, and after the style sheet SPEAK provides.

You can add multiple style sheets like this:

1. Create a folder and put your style sheets here.

2. Add a configuration item based on the Page-Stylesheet-Folder item template to your PageSettings.

3. Specify the path to the folder in the Folder field of the configuration item. It must be a fully qualified path, for example: `/sitecore/shell/client/MyApp/MyStyles`.

The files in the folder will be included in alphabetical order, based on their names. All files will be included after the style sheet SPEAK provides.

5.18.2   Page Settings

You define Page Settings like this:

- The Page Settings item must be an immediate child item of the page definition item.

- The Page Settings item must have the name "PageSettings".

- You can base a Page Settings item on any template.

- If you use the Speak-Page Settings template, your item will automatically have the correct name.

5.18.3   Properties

Name	Description	Default
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False

VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
PageCodeScriptFileName	Specify the script file for the page. The file must be named `<name of definition item>.js`. If, for example, the definition item is called `MyApp`, the page code script file must be called `MyApp.js`.	
PageCodeTypeName	Specify the type name and assembly of the Page Code object to be run server-side when the page is loading.	
SpeakCoreVersion	Specify which version of the SPEAK core to use: Speak Core 1-2: core version 1.2 Speak Core 1-1: core version 1.1	When not set, SPEAK core 1.1 is used.

5.18.4   Configuration

You can create a configuration item for a PageCode component by creating an item based on the PageCode Parameters template.

The configuration item has these specific fields:

- PageCodeScriptFileName
- PageCodeTypeName

5.19   ProgressIndicator

A ProgressIndicator control shows the user that a component in a SPEAK page has made a request and is waiting for a response. It does this by showing an animated graphic. You can specify if this graphic overlays only a specific control or the whole page. Users cannot interact with controls that are overlaid by a ProgressIndicator.

The ProgressIndicator is often used when a SPEAK component interacts with the backend server. For example, you often create "search pages" with this pattern:

- You use a SearchDataSource component to retrieve items

- You use a ListControl to show the items

If you want to show users when a search is in progress, you can do this by:

- Adding a ProgressIndicator to your page

- Setting the TargetControl of the ProgressIndicator to the ListControl

- Binding the IsBusy property of the ProgressIndicator to the IsBusy member of the SearchDataSource

5.19.1   Properties

Name	Description	Default
Appearance		
Behaviors	Behaviors extend the default behavior of components. You activate a behavior by entering the name, and you can activate multiple behaviors as a space-separated list.	
Delay	Specify the delay in milliseconds before the ProgressIndicator appears.	400 ms
IsBusy	If true, SPEAK shows the ProgressIndicator.	False
IsFullscreen	Is true, the ProgressIndicator covers the whole SPEAK page.	False
IsVisible	Use IsBusy instead.	True
ToolTip	Text displayed when the user hovers the mouse over the control. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	

	Use a configuration item based on the ProgressIndicator Parameters template to specify parameters for the ProgressIndicator control. You can specify the text and tooltip that the control displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
AutoShow	If True, SPEAK shows the ProgressIndicator automatically when a UI element makes a request and this request takes longer than a specific amount of time to conclude.	
AutoShowTimeout	Specify an amount of time (in milliseconds) that SPEAK waits before it shows the ProgressIndicator. SPEAK only shows the ProgressIndicator if AutoShow is True.	
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

5.19.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
targetControl	
isBusy	
isFullscreen	
delay	
width	
height	
left	
position	
top	

5.19.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
_getHeight	
_getLeft	
_getPosition	
_getTargetDomElement	
_getTop	
_getWidth	
_resizeHandler	
_setAutoShowMode	
_stopTimer	
_updateModel	
addBehavior	
changeBusy	
focus	
hide	

Name	Description
listenTo	
listenToOnce	
once	
set	
setBusy	
show	
stopListening	
toggle	

5.19.4 Configuration

You can create a configuration item for a ProgressIndicator control by creating an item based on the ProgressIndicator template.

The configuration item has these specific fields:

- IsFullscreen
- Delay
- IsBusy
- TargetControl
- AutoShow
- AutoShowTimeout

It inherits these fields:

- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id

5.20   RadioButton

A RadioButton control shows a standard radio button that users can select. You use a RadioButton as part of a group of RadioButtons. Only one RadioButton in a group can be selected at a time, and when a user selects a new RadioButton in a group, SPEAK automatically clears the previously selected button. You create a group by setting the GroupName of a set of RadioButtons to the same value. You must place buttons that are in the same group right after each other. It is technically possible not to do this, but this creates a confusing user experience.

You can set the Text property to create a label for a RadioButton. You can set the Value property to any string, and use this string in PageCode or a rule. The RadioButton always returns this string as the value of the Value property, also if the button is not selected.

When SPEAK first loads a page with a group of RadioButtons, and if you have not set IsChecked to true for any of the buttons in the group, the user sees a group with no selected buttons. When the user selects one of the buttons, it becomes impossible for the user to go back to the state where no button is selected. You can change this behavior by selecting a button with IsChecked when you design the page.

5.20.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsEnabled	If false, the control does not respond to clicks.	True
IsVisible	If false, the control does not take up space on the page.	True
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the RadioButton Parameters template, and point to this item here. You can specify the text and tooltip that the button displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	

PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
GroupName	Specify the name of the group this RadioButton is part of.	
IsChecked	If true, the RadioButton is selected.	False
Text	Specify the text that the RadioButton shows as a label.	
Value	Specify the text is the value of the RadioButton.	
Design		
Id	Lets you set the ID of the control.	

5.20.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
isEnabled	

Name	Additional Description
name	
value	
text	
isChecked	

5.20.3   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
_changeStatus	
_setGlobalValue	
addBehavior	
disable	
enable	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
show	
stopListening	
toggle	

5.20.4   Configuration

You can create a configuration item for a RadioButton by creating an item based on the RadioButton Parameters template.

The configuration item has these specific fields:

- IsChecked
- Value
- Text
- GroupName

It inherits these fields:

- IsEnabled

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

5.21 Separator

The Separator control shows a horizontal rule. You cannot set the size or the style.

5.21.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control and the controls it contains are visible.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False

VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

5.21.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	

5.21.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
show	
stopListening	
toggle	

5.21.4 Configuration

You can create a configuration item for a Separator by creating an item based on the Separator Parameters template.

The configuration item inherits these fields:

- IsVisible
- AccessKey
- Tooltip
- Behaviors
- Id

5.22   Slider

The Slider control shows a horizontal slider. Users can select a single value or a range of values by dragging the handle (or the handles) the control shows.

You can specify if the Slider control is a Value type (users can select one value) or a Range type (users can select the start and end value of a range of values.) The Slider can work either with a set of numbers or with a set of items containing a display name and a value.

You can also specify the following:

- The minimum and maximum selectable values

- The number of selectable values ("steps") between the minimum and the maximum

- The initially selected value or value range

- The color and size of the number

- The size of the image

5.22.1   Binding Items to the Slider control

You can bind items to the Slider control instead of setting numbers. If you create a number of items in a folder under PageSettings, you point to that folder in the DataSource property. You specify what field of these items the Slider control uses for the value or values in the DisplayFieldName property. It is these values the control makes available in the SelectedValue or the SelectedValueStart and SelectedValueEnd fields.

You specify what field of these items the Slider control uses for the tooltips that it shows when users drag a handle.

5.22.2   Rebind

You use the `rebind` method to change the data handled by the control programmatically.

You can also bind items programmatically to the slider by calling the public method rebind(items) on the slider component, like this:

```
var newItems = [{ "itemId": "{17E6304B-ED32-4A96-A0A1-5C31E5C309A2}", "itemName": "xxVery Bad",
"value": "100" }, { "itemId": "{02B5F74D-1C86-4658-B71C-F0D20A367F4D}", "itemName": "xxBad",
"value": "200" }, { "itemId": "{44E1BBDA-92DF-4673-9A90-FD02564A92EC}", "itemName": "xxNormal",
"value": "300" }];

this.Slider.rebind(newItems);
```

You pass the items as a JSON array of objects of the type: `{itemId, itemName, value}`. You can only set the ValueFieldName and DisplayFieldName properties only when new data is passed to the control.

`rebind(null)` is a special case. It will make Slider control that was bound unbound.

When you pass in new data (or set the control to unbound) you have to consider the setting of the `Minimum`, `Maximum`, and `Step` properties. If they are not set, the previous values will remainif effect.

If you call `rebind()` without the Items parameter, it makes property changes effective for the current data the control handles. Note that ValueFieldName and DisplayFieldName properties are exceptions to this.

5.22.3 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
DisplayFieldName	Specify the name of the data field that the control uses to display as a tooltip for the selected item.	_Display name
IsEnabled	If false, the control is disabled.	True
IsVisible	If false, the control does not take up any space on the page.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Type	Specify the type of the Slider control. You can select Single or Range.	Single
ValueFieldName	Specify the name of the data field of the selected item the control uses for the values of SelectedValue, SelectedValueStart and SelectedValueEnd.	
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource		
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False

VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Items		
Maximum	Specify the maximum value that users can select in the Slider control. Maximum and Mininum together specify the scale of the Slider control.	
Minimum	Specify the minimum value that users can select in the Slider control. Maximum and Mininum together specify the scale of the Slider control.	
SelectedValue	Specify the selected value of a Slider control of type Single.	Maximum
SelectedValueEnd	Specify the value selected as the upper end of a range of values. This is for a Slider control of type Range.	Maximum
SelectedValueStart	Specify the value selected as the lower end of a range of values. This is for a Slider control of type Range.	Minimum
Step	Specify the number of discrete value the control makes it possible to select between the maximu and the minimum in a Slider control of type Range.	
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

5.23 SubAppRenderer

You use a SubAppRenderer to include one SPEAK page in the layout of another SPEAK page. For example, you can use the SubAppRenderer to show a menu that you reuse in different SPEAK pages.

You use the SubAppRenderer like this:

1. Create a common SPEAK page that you include in other SPEAK pages. Use the Speak-DialogPage template and the Speak-EmptyLayout.

2. Add SPEAK components to the layout. Do not specify placeholders.

3. Do not add PageCode or structure components to this common page.

4. Add a SubAppRenderer component to each page where you want to include the common page you just created.

5. Point to the common page in the DataSource property of the SubAppRenderer.

Note that SPEAK renders all controls from the included common page in the placeholder of the SubAppRenderer. If this is not what you want, you have to create multiple common pages and include these pages with multiple SubAppRenderer controls.

5.23.1 Properties

Name	Description	Default
DataSource	Specify the ID of the SPEAK page the SubAppRenderer control renders.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	The SupAppRenderer control renders all controls in the page you specify in DataSource in the placeholder.you specify here.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on	False

	rendering parameters passed to the presentation component.	
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False

5.24 SubPageCode

The SubPageCode component is similar to the PageCode component, but some features are different. You use the SubPageCode to add code that is specific for an item. Sitecore only executes this code when it renders the item.

It differs from the PageCode component in these ways:

- You use the SubPageCode component for items that are rendered by a Repeater component.

- You add the SubPageCode component to the _Standard Values layout of the template for the items that the Repeater component renders.

5.24.1 Properties

Name	Description	Default
DataSource	Points to a configuration item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID. You do not need to change this.	Page.Body
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False

Data		
PageCodeScriptFileName	Specify the folder where the script files for the item are located.	
PageCodeTypeName	Specify the type name and assembly of the Page Code object to be run server-side when the page is loading.	

5.24.2 Configuration

You can create a configuration item for a SubPageCode component by creating an item based on the SubPageCode Parameters template.

The configuration item has these specific fields:

- PageCodeScriptFileName
- PageCodeTypeName

5.25 Text

A Text control shows a text string. The TextType property allows you to choose between a number of predefined styles for the string.

5.25.1 Properties

Name	Description	Default
Appearance		
Behaviors	Behaviors extend the default behavior of components. You activate a behavior by entering the name, and you can activate multiple behaviors as a space-separated list.	
IsVisible	When false, the Text control does not take up space on the page.	
Text	Text displayed by the Text control. You can enter a static text, set the text from PageCode, or you can bind to a property of another control. You should not specify a static text in the component properties if your dialog or application supports multiple languages. See the DataSource property for an alternative way to specify text.	
TextType	Specifies how the control formats the text. The following text types are available: Text, Label, LargeLabel, Value, LargeValue, Title, LargeTitle. SmallTitle, and Divider.	Text
ToolTip	Text displayed when the user hovers the mouse over the Text control. You should not specify a static text in the component properties if your dialog or application supports multiple languages. See the DataSource property for an alternative way to specify text.	None
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute, a single character.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the Text Parameters template to specify parameters for the Text control.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Text Id of the placeholder where Sitecore renders this control.	

Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN.	False
VaryByData	Controls whether output caching varies based on the data source of the control.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Data		
FieldName		
Design		
Id	Lets you set the ID of the control.	

5.25.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
text	

5.25.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
focus	
hide	
set	
show	
toggle	

5.25.4 Configuration

You can create a configuration item for a Text control by creating an item based on the Text Parameters template.

The configuration item has these specific fields:

- TextType

- Text

- Fieldname

It inherits these fields:

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

5.26   TextArea

A TextArea control is a text box that users can enter input in. Users can enter multiple lines of text (as opposed to the TextBox control that is one-line only).

You can set a watermark or a default text that the TextArea displays. A watermark is a non-editable text that the TextArea control shows when it is empty. You can enter a text that helps users to understand what they can enter. The watermark disappears as soon as the user enters a character, and it reappears if the user deletes all previously entered characters.

A default text is a text that the control displays, but users can change and delete this text. If you specify both a watermark and a default text, the TextArea control initially shows the default text, but if the user deletes all of this text, the watermark appears.

You can specify rows and columns to size the TextArea control. If you do not specify any of these, SPEAK gives the control a default size. The rows and columns are not absolute dimensions. They correspond directly to the Rows and Columns attributes in HTML.

Some browsers make it possible for users to resize the TextArea. These browsers display a resizing handle in the bottom-right corner of the TextArea. The control cannot be resized smaller than the size it was initially. Other controls can move around when users resize the TextArea.

5.26.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
Columns	Specify the width of the control in columns. The "columns" are HTML columns.	
IsEnabled	If false, users cannot enter text in the control.	True
IsVisible	If false, the control does not take up space on the page.	True
Rows	Specify the height of the control in rows. The "rows" are HTML rows.	
Text	Specify a default text that the control shows. Users can edit and delete this text.	
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Watermark	Specify a text that the control shows when it has no input from the user. The text is not editable, and it is not shown when there is any user input.	
WidthPolicy	Specify how SPEAK interprets the width of the control. You can specify: • Default: the default width of the control is used. If you specify a value in Columns, SPEAK uses this value.	Default.

	• BlockLevel: the width is set relative to the width of the container. A value set in Columns is disregarded. The TextArea control fills the container horizontally.	
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the TextArea Parameters template, and point to this item here.	
MaxLength	Specify the maximum number of characters users can input in the TextArea control. The control does not accept any input when this number has been reached.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Wrap	Specify how the control wraps input lines that are longer than the width of the control: soft: Text wraps, but no return and line feed characters are inserted. hard: Text wraps, and return and line feed characters are inserted. off: the text does not wrap. Some browsers display a horizontal scrollbar when the length of the line exceeds the width of the TextArea.	soft
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False

VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

5.26.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
isEnabled	
text	
isReadOnly	
cols	
maxLength	
rows	
wrap	
isRequired	
watermark	

5.26.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
disable	
enable	
focus	
hide	
listenTo	
listenToOnce	

Name	Description
once	
set	
show	
stopListening	
toggle	

5.26.4 Configuration

You can create a configuration item for a TextArea control by creating an item based on the TextArea Parameters template.

The configuration item has these specific fields:

- Columns

- Rows

- Watermark

- Text

It inherits these fields:

- IsEnabled

- IsVisible

- AccessKey

- Tooltip

- Behaviors

- Id

5.27 TextBox

A TextBox control is a text box that users can enter input in. Users can enter one line of text (as opposed to the TextArea control where users can enter multiple lines of text).

You can set a watermark or a default text that the TextBox displays. A watermark is a non-editable text that the TextBox control shows when it is empty. You can enter a text that helps users to understand what they can enter. The watermark disappears as soon as the user enters a character, and it reappears if the user deletes all previously entered characters.

A default text is a text that the control displays, but users can change and delete this text. If you specify both a watermark and a default text, the TextBox control initially shows the default text, but if the user deletes all of this text, the watermark appears.

You can mark a TextBox as required. Some browsers style a TextBox that is marked this way in a different way.

If you set the Type property to "password," the TextBox masks user input. Most browsers replace the characters that the user types with * (asterix) characters.

5.27.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsEnabled	If false, users cannot enter text in the control.	True
IsVisible	If false, the control does not take up space on the page.	True
Text	Specify a default text that the control shows. Users can edit and delete this text.	
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Type	Specify either "text" or "password".	text
Watermark	Specify a text that the control shows when it has no input from the user. The text is not editable, and it is not shown when there is any user input.	
WidthPolicy	Specify how SPEAK interprets the width of the control. You can specify: • Default: the default width of the control is used. • BlockLevel: the width is set relative to the width of the container. The control fills the container horizontally.	Default.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	

DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the TextBox Parameters template, and point to this item here.	
IsReadonly	If True, the users cannot enter text.	False
IsRequired	If True, the TextBox control is marked as being "required". There is no logical effect, but some browsers style the text box differently.	False
MaxLength	Specify the maximum number of characters users can input in the TextBox control. The control does not accept any input when the number of characters in the input reach this number.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

5.27.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
isEnabled	
text	
isReadOnly	
isRequired	
maxLength	
watermark	

5.27.3   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
checkEnterKey	
disable	
enable	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
show	
stopListening	
toggle	

5.27.4   Configuration

You can create a configuration item for a TextBox control by creating an item based on the TextBox Parameters template.

The configuration item has these specific fields:

Text
Type
Watermark

It inherits these fields:

IsEnabled
IsVisible

AccessKey
Tooltip
Behaviors
Id

5.28   ToggleButton

A ToggleButton control presents a standard button that users can click. It is similar to a Button control, but adds the following functionality:

- A ToggleButton has two states, and when users click the ToggleButton, it toggles the state it is in. The ToggleButton control remembers what state it is in until the user reloads the page that contains the ToggleButton. The ToggleButton uses different visual styles for the two states.

- The IsOpen and the ShowArrow properties are specific for the ToggleButton. It shares all other properties with the Button control.

This is following is an example of when to use a ToggleButton control.

When you use the FilterControl to show a panel with facets, you often want to give users the possibility to hide and show the FilterControl. You can use a ToggleButton for this, by binding the IsVisible property of the FilterControl to the IsOpen property of the ToggleButton. The default value of IsOpen is false, so if you want the FilterControl to be open when the page is loaded, you have to set IsOpen to true.

If you set the ShowArrow property to true, the ToggleButton shows an arrow that points up when IsOpen is true, and points down when IsOpen is false.

If you set the ButtonType property to "primary," SPEAK makes it stand out visually. You can set the size of a button to "normal" or "large" with the Dimensions property.

You should set ToggleButtons that users click to carry out the main purpose of a page "primary" and use "default" for other buttons. You can also set the ButtonType to "inverse." An "inverse" button uses the same colors as a "default" button, but swaps the text and background colors.

ToggleButtons can display icons as well as text and they can show both at the same time.

You specify icons using the ImageUrl property. You can use the same icons here as when you select an icon for an item in the content tree. Button controls also support defining the icon as a "sprite," selected from a sprite sheet. The ImageUrl property stores the URL of the sprite sheet, and the BackgroundPosition property stores the coordinates of the desired sprite (icon/image).

If you use both icon and text, SPEAK positions the icon to the left of the text. You cannot change this position.

5.28.1   Properties

Name	Description	Default
Appearance		
BackgroundPosition	Lets you set the offset of the sprite/icon/image in a sprite sheet file in this format: -<x>px -<y>px Where <x> indicates the horizontal number of pixels from the left edge of the sprite sheet to the left edge of the sprite (represented as a negative number), and <y> indicates the vertical number of pixels from the top edge of the sprite sheet (represented as a negative number) to the top edge of the sprite.	-0px -0px

Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
ButtonType	You can choose between Default, Primary and Inverse.	Default
Dimension	Lets you set the size of the button. You can choose between Normal and Large.	Normal
ImageUrl	Lets you set the Sitecore path of an icon (for example, "Network/16x16/home.png" for the icon shown for the "/sitecore/content/home" item) or sprite sheet. SPEAK provides a number of sprite sheets in the "\sitecore\shell\client\Speak\Assets\img" folder (found in the Website folder).	
IsEnabled	If false, the Button does not respond to clicks.	True
IsOpen	If true, the ToggleButton is in the "open" state. The ToggleButton toggles the value when users click it.	False
IsVisible	If false, the Button does not take up space on the page.	True
ShowArrow	If true, the ToggleButton shows an arrow that indicates what state it is in.	False
Text	Lets you set the text you want displayed on the Button. You can enter a static text, set the text from PageCode, or you can bind to a property of another control. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the control. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the ToggleButton Parameters template, and point to this item here. You can specify the text and tooltip that the button displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	

PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true.	False
	Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

5.28.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
isEnabled	
text	
backgroundPosition	
isOpen	

5.28.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
click	
close	
disable	
enable	
focus	
hide	
listenTo	
listenToOnce	
once	
open	
set	
show	
stopListening	
toggle	
toogle	

5.28.4 Configuration

You can create a configuration item for a ToggleButton by creating an item based on the ToggleButton Parameters template.

The configuration item has these specific fields:

ButtonType
Dimension
Text – this field corresponds to the Text property, and you use the field to make it easier to translate UI texts.
IsOpen
ImageUrl
BackgroundPosition
ShowArrow

It inherits these fields:

Click
IsEnabled
IsVisible
AccessKey
Tooltip
Behaviors
Id

5.29   ToolTip

A ToolTip control shows a tooltip. A tooltip is a small, rectangular pop-up window that contains a brief description of a SPEAK control. All SPEAK controls have a ToolTip property that also shows a tooltip for the control. When you use the ToolTip control instead of the ToolTip property, you can set the style and the behavior of the pop-up window in a number of ways that you cannot do for the ToolTip property.

The ToolTip control shows a ToolTip for a specific control. You specify the association between the controls in the TargetControl property of the ToolTip control.

5.29.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the ToolTip does not take up space on the page.	True
Simple	If false, the ToolTip control shows the value of the Title property as a title.	True
ToolTip	Not useful for this control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
ContainsHTML	If true, HTML in Content and Title is rendered as HTML.- If false, HTML is shown literally.	False
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the ToolTip Parameters template, and point to this item here. You can specify the texts that the tooltip displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Placement	Specify how SPEAK positions the ToolTip relative to the control it is a ToolTip for. Possible values: bottom, left, right, top.	Bottom.

TargetControl	Specify the ID of the control that the ToolTip is a ToolTip for.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Content	Specify the text of the ToolTip. If ContainsHTML is true, the ToolTip control parses and renders HTML.	
Title	Specify a text that the ToolTip control shows as a title when the Simple property is "false".	
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

5.29.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "text" in the following table corresponds to the "Text" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
html	
placement	
content	

selector	
title	
trigger	
delay	
simple	
target	
type	
name	
isVisible	

5.29.3   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
show	
stopListening	
toggle	
updateView	

5.29.4   Configuration

You can create a configuration item for a ToolTip by creating an item based on the ToolTip Parameters template.

The configuration item has these specific fields:

Animation
Simple
Delay
TargetControl
Placement
ContainsHTML
Content
Title

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

Chapter 6

Containers

6.1 Accordion

An Accordion control is a container that has two parts: a header and a body. The body part has two states: open and closed. Users can toggle between the open and the closed state by clicking an icon that the control always shows. The Accordion always shows the header part.

The control has two placeholders:

- The ActionBar is the header part. You can add buttons, for example, to this part. The open/close button is always added to this part, and you cannot change the style of this button. You can create a heading (or a title) by using the Header property.

- The Body is the part that can open and close.

If you only need the open/close feature, and not the ActionBar, you should use an Expander control instead.

You can use the EnableAddtional/ShowAdditional properties to show and hide controls that are inside the Accordion.

6.1.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
ContentHeight	Lets you specify the height of the control. You can enter an absolute height, or a percentage.	100%
EnableAdditional	If true, the control shows "More/Less" links.	False
Header	Specify a text that the control shows in the ActionBar.	
ImageUrl	Lets you set the Sitecore path of an icon that the Accordion shows to the left of the Header.	
IsOpen	If true, the control and the controls it contains are open.	True
IsVisible	If true, the control and the controls it contains are visible.	True
ShowAdditional	This property is True when EnableAdditional is True and the user has clicked "More", and it is False when the user has clicked "Less". ShowAdditional is False when EnableAdditional is False or undefined. You bind to ShowAdditional from the IsVisible property of other controls in the Accordion. These controls toggle between being visible and invisible when the user clicks the "More" and "Less" links provided by the EnableAddtional property.	False
ToolTip	The Border control displays this text when users hover the	None.

	mouse over the control.	
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

6.1.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming

convention.

Name	Additional Description
isVisible	
text	
isOpen	
showAdditional	
enableAdditional	
contentHeight	
isVisible	

6.1.3   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
close	
focus	
hide	
hidePromotedFields	
listenTo	
listenToOnce	
once	
open	
set	
show	
showPromotedFields	
stopListening	
toggle	
toggleAdditional	

6.1.4   Configuration

You can create a configuration item for an Accordion control by creating an item based on the Accordion Parameters template.

The configuration item has these specific fields:

Header
IsOpen
EnableAdditional
ShowAdditional
ContentHeight

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

6.2   Border

The Border control contains other controls. It creates a placeholder called Content when you add it to a page, and the Border control contains any control that has this placeholder as the PlaceholderKey.

A Border control can have a border and padding. You can align the controls it contains in different ways.

You cannot set the size of a Border control; it has no specific size set. You can use a Border control that contains no controls to produce an empty HTML "div" element.

6.2.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
ContentAlign	Lets you specify the alignment of the contained controls: Left, Center, or Right.	Left
Height	Specify the height of the control in pixels.	Null
IsVisible	If true, the Border control and the controls it contains are visible.	True
ShowBorder	If true, the Border control has a border.	False
ToolTip	The Border control displays this text when users hover the mouse over the control.	None.
UsePadding	If true, the Border control has padding. The padding is the extra space between content and border — it does not matter if the border is visible or not.	False
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true.	False

	Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

6.2.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
width	
height	

6.2.3   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
focus	
hide	
set	
show	
toggle	

6.2.4 Configuration

You can create a configuration item for a Border control by creating an item based on the Border Parameters template.

The configuration item has these specific fields:

ContentAlignment
ShowBorder
UsePadding

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

6.3　ColumnPanel

A ColumnPanel control is a container you use when you create a fluid grid. You add ColumnPanels to a RowPanel container: one for each column you want in your layout. You then add controls that show actual content to the ColumnPanel containers.

SPEAK provides an underlying grid with 12 columns. You can specify how wide your ColumnPanel is in "grid columns." You should not add ColumnPanels where the sum of "grid columns" is more than 12.

You can use ColumnPanels with no content – SPEAK shows such columns, but they are left empty.

6.3.1　Properties

Name	Description	Default
Appearance		
Behaviors	Behaviors extend the default behavior of components. You activate a behavior by entering the name, and you can activate multiple behaviors as a space-separated list.	
ContentAlign	Specifies the align of contained controls within the column created by the ColumnPanel control: Left, Right or Center.	Left.
GridColumns	Specifies the number of "grid columns" this ColumnPanel use. Can be from 1 to 12.	
IsVisible	If true, the control and the control it contains are visible.	True
ToolTip	Text displayed when the user hovers the mouse over the control.	None.
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute. This is a single character.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Text Id of the placeholder where Sitecore renders this control.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the	False

	HTML generated. For more information regarding caching, please refer to the SDN.	
VaryByData	Controls whether output caching varies based on the data source of the control.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

6.3.2 Configuration

You can create a configuration item for a ColumnPanel by creating an item based on the ColumnPanel Parameters template.

The configuration item has these specific fields:

GridColumns
ContentAlign

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

6.4 Expander

An Expander control is a container that can open and close. Users can toggle between the open and the closed state by clicking an icon that the control always shows. An Expander control opens or closes with a brief sliding animation. You can add a text (a "title") that is shown in both the open and in the closed state by setting the Text property.

You add components to an Expander control by setting the PlaceholderKeys to the Expander control's Body placeholder.

The default state of an Expander is "open."

The Accordion control offers similar features and, for example, adds the capability to have buttons in a "header" that is always open.

6.4.1 Properties

Name	Description	Default
Appearance		
Behaviors	Behaviors extend the default behavior of components. You activate a behavior by entering the name, and you can activate multiple behaviors as a space-separated list.	
IsOpen	If true, the Expander control displays its contents. If false, the component appears collapsed.	True
IsVisible	If true, the Expander control and the controls it contains are visible.	True
Text	Lets you specify a text that is always shown (a "title"). You can enter a static text, set the text from PageCode, or you can bind to a property of another control. If your dialog or application supports multiple languages, do not specify a static text in the component properties . See the DataSource property for an alternative way to specify text.	
ToolTip	Text displayed when the user hovers the mouse over the control. If your dialog or application supports multiple languages, do not specify a static text in the component properties . See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the Expander Parameters template to specify parameters for the Expander control. You	

	can specify the text and tooltip that the control displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

6.4.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
width	
Height	
isOpen	

6.4.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
Focus	
Hide	
Set	
show	
toggle	.
toogle	

6.4.4 Configuration

You can create a configuration item for an Expander control by creating an item based on the Expander template.

The configuration item has these specific fields:

IsOpen
Text - this field corresponds to the Text property, and you use the field to make it easier to translate UI texts.

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

6.5 Frame

A Frame control renders a standard HTML inline frame ("iframe"). The SourceUrl property defines the address of the shown HTML document.

Note

The browser settings of the user can interfere with the content that the control shows (for example, the cross-site scripting settings may disable features required by the referenced HTML page).

6.5.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the Frame does not take up space on the page.	True
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the Frame Parameters template, and point to this item here. You can specify the tooltip that the Frame displays in the configuration item. It is much easier to translate the UI texts of the dialog or the application when you use this approach.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	

Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Height	Lets you set the height of the Frame control. You can use percentages ("100%") or pixels ("600"). If you use a percentage, SPEAK sets the height of the Frame control so that it fills the available vertical space of the containing element in the layout. You must be very careful if you use pixels because the layout SPEAK creates is fluid and it does not have absolute dimensions.	
SourceUrl	Lets you set the URL of the HTML document to show, for example: "http://www.sitecore.net".	
Width	Lets you set the width of the Frame control. You can use percentages ("100%") or pixels ("600"). If you use a percentage, SPEAK sets the width of the Frame control so that it fills the available vertical space of the containing element in the layout. You must be very careful if you use pixels because the layout SPEAK creates is fluid and it does not have absolute dimensions.	
Design		
ID	Lets you set the ID of the control. You should use a semantic name (for example: NewsFrame).	

6.5.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
width	
height	
sourceUrl	

6.5.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
focus	
hide	
set	
show	
toggle	

6.5.4 Configuration

You can create a configuration item for a Frame control by creating an item based on the Frame Parameters template.

The configuration item has these specific fields:

SourceUrl
Width
Height
IsDeferred

6.6   LoadOnDemandPanel

You use the LoadOnDemandPanel control to load and render an item. The item is not loaded and rendxered until you call the **load** member.

The LoadOnDemandPanel control provides a number of triggers you can listen for in you code:

- **beforeSend**
  The control raises this event before it makes an Ajax request. You can catch this event and write code that modifies the XMLHttpRequest object and sets additional headers, if you need to.

- **success**
  The control raises this event if the Ajax request was successful (that is: the control did not receive errors from the server, and it did not detect errors in the data).

- **error**
  The control raises this event if the Ajax request did not succeed. **success** and **error** are mutually exclusive.

- **complete**
  The control always raises this event, for both succesul and for failed request and even for synchronous requests.

You use the LoadOnDemandPanel to not let SPEAK load an item immediately. You can tell SPEAK to load the item in code. For example, you can tell SPEAK to load the item when a specific condition is met. You can also tell SPEAK to refresh (reload) and item it has previously loaded.

Properties

Name	Description	Default	Availability
Appearance			
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.		
Behavior			
AccessKey	Lets you specify a single character that SPEAK will render as the HTML5 AccessKey attribute of the control.		
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.		
Parameters	Text string with name-value pairs. An alternative method for assigning property values.		
Behavior			
ItemId	Point to the item you want the control to load.		Client, Server
Database	Specify the database the control loads the item from. You can use the $context_database or $context_contentdatabase tokens, or you can name the database explicitly.		Client, Server
TriggerLoad	You use this property to set the client-side event from the server.		Server

IsLoaded	This read-only property is true when the control has loaded the item.		Client(readonly), Server
LoadOnVisible	If this property is true, the control loads content when the IsVisible property is true.	False	Server
Caching			
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False	
VaryByData	Lets you specify whether or not output caching varies based on the data source of the control.	False	
VaryByDevice	Lets you specify whether or not caching varies based on the name of the context device.	False	
VaryByLogin	Lets you specify whether or not output caching varies based on whether or not the user has authenticated.	False	
VaryByParameters	Lets you specify whether or not output caching varies based on rendering parameters passed to the presentation component.	False	
VaryByQueryString	Lets you specify whether or not output caching varies based on query string parameters passed in the URL.	False	
VaryByUser	Lets you specify whether or not output caching varies by the domain and username of the context user.	False	
Design			
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).		

6.6.1   Model Members – client-side properties

Model members represent the client-side properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation: "itemId" in the following table corresponds to the "ItemId" server-side property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
itemId	
database	

6.6.2   View Members – client side events

View members represent events that the component either raises or listens to. You can listen for and raise events in the PageCode.

Name	Description
beforeSend	The control raises this event before it makes an Ajax request.
success	The control raises this event when it has made a successful request.
error	The control raises this event if it has received an error as response to a request.
complete	This event is called regardless of if the request was successful, or not. You will always receive a complete callback, even for synchronous requests.

6.6.3   Client-side methods

Client-side methods represent the list of methods exposed by control.

Name	Description
load	This methodmloads the content and sets the isLoaded property to true. If the content is already loaded, it will do nothing.
destroy	This method destroys the panel and all nested controls.
refresh	This method resets the isLoaded property and call load method.

6.7   ProtectedBorder

The ProtectedBorder control is very similar to the Border control. Itcontains other controls. It creates a placeholder called Content when you add it to a page, and the ProtectedBorder control contains any control that has this placeholder as the PlaceholderKey.

It has two properties that the Border control does have: RequiresRead and RequiresWrite. When you use these properties, the ProtectedBorder control only renders the controls it contains if the context user has specified access rights.

A ProtectedBorder control can have a border and padding. You can align the controls it contains in different ways.

You cannot set the size of a ProtectedBorder control; it has no specific size set. You can use a ProtectedBorder control that contains no controls to produce an empty HTML "div" element.

6.7.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
ContentAlign	Lets you specify the alignment of the contained controls: Left, Center, or Right.	Left
Height	Specify the height of the control in pixels.	Null
IsVisible	If true, the ProtectedBorder control and the controls it contains are visible.	True
RequiresRead	If true, the ProtectedBorder control only renders the controls it contains if the context user has Read rights on the data source item.	False
RequiresWrite	If true, the ProtectedBorder control only renders the controls it contains if the context user has Write rights on the data source item.	False
ShowBorder	If true, the ProtectedBorder control has a border.	False
ToolTip	The ProtecteBorder control displays this text when users hover the mouse over the control.	None.
UsePadding	If true, the ProtecteBorder control has padding. The padding is the extra space between content and border — it does not matter if the border is visible or not.	False
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the	

	configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

6.7.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
width	
height	

6.7.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
focus	
hide	
set	
show	
toggle	

6.7.4 Configuration

You can create a configuration item for a Border control by creating an item based on the ProtecteBorder Parameters template.

The configuration item has these specific fields:

ContentAlignment
ShowBorder
UsePadding

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

6.8 Repeater

The Repeater control is a container. You bind the Repeater control to a data source, and the Repeater control shows the items that the data source provides.

The Repeater is different from other containers (such as the ListControl) in these ways:

- The Repeater control does not render items.

- You can add and remove controls that show items client-side. This means that SPEAK does not have to reload the page when items are added or removed.

You use the Repeater control this way:

1. Create a template in the core database for the items you want the Repeater control to show.

2. Open the Design Layout of the _Standard Values of the template you created.

3. Add renderings to this layout.

4. Create items based on this template in the core database. (You can use the Item Web API or Sitecore.Services.Client to create items if you need to do this programmatically.)

5. Add a data source component to you page. This data source component must return the items you created.

6. Bind the Items property of the Repeater control to the data source.

The Repeater control shows the items with withn the renderings you added to _Standard Values.

The Repeater itself does not provide logic for adding and removing items, but you can add this functionality in PageCode.

6.8.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control and the controls it contains are visible.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	

PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Items	Points to the items the control shows. You can use the dropdown to select a binding to another control. You typically select a binding to the items property of a data source component.	
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

6.8.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
items	
isLoading	

6.8.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
add	
addBehavior	
focus	
hide	
listenTo	
listenToOnce	
once	
renderItem	
renderItems	
reset	
set	
show	
stopListening	
toggle	

6.8.4 Configuration

You can create a configuration item for a Repeater by creating an item based on the Repeater Parameters template.

The configuration item has these specific fields:

Items

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

6.9 RowPanel

A RowPanel control is a container that you use when you create a fluid grid. You use the RowPanel control to start a new row in the grid, and then you add ColumnPanels controls to the RowPanel. You add the controls that show the actual content to the ColumnPanels.

6.9.1 Properties

Name	Description	Default
Appearance		
Behaviors	Behaviors extend the default behavior of components. You activate a behavior by entering the name, and you can activate multiple behaviors as a space-separated list.	
IsVisible	If true, the control and the control it contains are visible.	True
ToolTip	Text displayed when the user hovers the mouse over the control.	None.
UsePadding	If true, the control uses padding. Padding is extra space between content and the "border".	False
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute. This is a single character.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Text Id of the placeholder where Sitecore renders this control.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN.	False
VaryByData	Controls whether output caching varies based on the data source of the control.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False

VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

6.9.2 Configuration

You can create a configuration item for a RowPanel control by creating an item based on the RowPanel Parameters template.

The configuration item has this field:

UsePadding

6.10 Section

A Section control adds an HTML5 section and wraps all the controls it contains in <section>…..</section> tags.

6.10.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control and the controls it contains are visible.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False

VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

6.11   SmartPanel

A SmartPanel control contains a set of components in a panel that slides out from the edge of the browser in response to the open event or when the IsOpen property is set to true and slides back to the hidden state in response to the close event or when the IsOpen property is set to false. You can use the same controls in a SmartPanel as you can on a page. When the SmartPanel opens, it covers a portion of the page. You add both the SmartPanel and the components contained in the SmartPanel in the page layout. You position components in the SmartPanel using the SmartPanel's placeholder keys. You can access all components in the SmartPanel from the PageCode of the page that hosts the SmartPanel.

A SmartPanel always shows a Close "icon button" on the right side of its Header placeholder. You cannot access properties and members of this button, and you cannot remove it. The SmartPanel closes when a user clicks this button.

You can select between three predefined sizes of the SmartPanel, and you can specify from where the SmartPanel appears when it opens.

6.11.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
Dimensions	Specifies the size of the panel. You select between three predefined sizes: Normal, Small and Large.	Normal
IsOpen	Opens or closes the SmartPanel. You normally use this from PageCode to open and close the SmartPanel.	False.
IsVisible	Is not used by this control.	
OffsetBottom	Lets you specify the number of pixels the SmartPanel is offset from the bottom of the page.	0
OffsetTop	Lets you specify the number of pixels the SmartPanel is offset from the top of the page.	0
Position	Lets you specify the position where the panel appears. You can choose between Top, Bottom, Left and Right.	Right
ToolTip	Text displayed when the user hovers the mouse over the SmartPanel.	None.
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute. This is a single character.	
DataSource	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the	

	DataSource property for an alternative way to specify text.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID. You do not need to change this property.	Page.Body
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

6.11.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isOpen" in the following table corresponds to the "IsOpen" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
width	
height	
isOpen	
dimensions	
position	
offsetTop	
offsetBottom	

6.11.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
_changeStatus	
close	
collapse	
expand	
focus	
hide	
open	
set	
show	
toggle	

6.11.4 Configuration

You can create a configuration item for a SmartPanel control by creating an item based on the SmartPanel Parameters template.

The configuration item has these specific fields:

Position
Dimensions
OffsetTop
OffsetBottom
IsOpen

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

6.12 TabControl

A TabControl displays one or more tabs.

You define the tabs of a TabControl with configuration items based on the Tab template ("tab configuration items"). You add content to these tabs by adding components to the tab configuration items. By convention, you create the tab configuration items in a folder under PageSettings.

You can add the tabs to the TabControl in three ways:

- Put the tab configuration items in a folder and point the Data Source property of the TabControl to this folder. The TabControl shows the tabs in the order that they appear in the folder. The TabControl always shows *all* the tab items in this folder.

- Add the tab configuration item GUIDs to the Tabs property of the TabControl in a pipe ('|') separated list. The TabControl shows the tabs in the order that you list them.

- Add the tabs dynamically from code. You create a list of tab items and add this list to the "tabs" member of the TabControl. The TabControl shows the tabs in the order that are in the list.

A TabControl always uses the DisplayNames of the tab configuration items as the tab labels.

Tab configuration items can include any component. You do not need to specify a placeholder for components that you add to the tab. Components defined on the tab configuration item without placeholders appear in the tab content area. If you add a component that introduces placeholders to the tab, you can assign components to these placeholders as you would normally.

You can add a TabControl to a tab configuration item layout to add a set of ("nested") tabs within a tab. One warning: you should not add a folder for the nested tab items subfolder of a folder that contains tab items (folders are read recursively, so the tabs items would get mixed up).

You normally set the TabType property of the nested TabControl to "SubTab" to create the best visual experience.

The TabType property defines the style of the tab buttons. Valid options include:

- **Normal**
  The individual tabs appear as rectangular buttons with a background color.

- **Progressive**
  The individual tabs appear as "arrow" buttons with a background color. You can use this to give users a visual clue about the steps in a sequence.

  The TabControl itself does not provide any logic for the sequence of steps. You can use the SelectedTab property/member and the SelectedTabIndex member to implement this logic in the PageCode.

- **SubTab**
  The individual tabs have a different appearance, which makes it easier to differentiate the enclosed TabControl from the parent tab. This option is useful when you include a TabControl in a tab.

6.12.1 Properties

Name	Description	Default
Appearance		
Behaviors	Behaviors extend the default behavior of components. You activate a behavior by entering the name, and you can activate multiple behaviors as a space-separated list.	
IsVisible	If true, the Frame is visible.	True
TabType	Set to Normal, Progressive or SubTab.	Normal
ToolTip	Text displayed when the user hovers the mouse over the control.	None
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute. This is a single character.	
DataSource	Point to a folder that contains tab configuration items. The TabControl shows each of these tabs in the order that they appear in the folder.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Text Id of the placeholder where Sitecore renders this control.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN.	False
VaryByData	Controls whether output caching varies based on the data source of the control.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False

Data		
SelectedTab	The Item ID of the selected tab. You can read and write this property.	TabControl selects the first tab (the first in the folder or in the list) initially if this property is not set.
Tabs	A list of Item IDs of tabs.	
Design		
Id	Lets you set the ID of the control.	

6.12.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
isVisible	
width	The rendered width in pixels (read-only).
height	The rendered height in pixels (read-only).
staticTabs	
dynamicTabs	
selectedTab	
isProgressive	
tabs	
selectedTabIndex	The index of the selected tab in the list of tabs (starting from 0).

6.12.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addDynamicTab	
focus	
hide	
onTabClicked	
set	
show	
toggle	

6.12.4 Configuration

You can create a configuration item for a TabControl by creating an item based on the TabControl Parameters template.

The configuration item has these specific fields:

Tabs
SelectedTab
TabType

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

Chapter 7

Data

7.1 ChartDataProvider

You use the ChartDataProvider component to provide data for the Chart controls.

The ChartDataProvider provides data by making a request to an MVC Controller. You specify the URL of the MVC Controller on the server, and you can specify query parameters. You can also set filters on date range and time resolution. When you set a filter on time resolution, the date will grouped according to the filter (for example, day, week, or month).

The ChartDataProvider does not retrieve data from the MVC Controller immediately. You have to tell the ChartDataProvider to call the MVC Controller.

You use the `getData` function to ask the MVC Controller for data, for example like this in PageCode:

```
        var requestOptions =
         {
           parameters: "",
           onSuccess: this.getDataCallback,
           url: "/api/sitecore/speakcharting/chart/Data"
         };
    …
        this.ChartDataProvider.viewModel.getData(requestOptions);
```

"url" can be set as a property as well (DataUrl). If you set it both in code and as a property, the ChartDataProvider will use the value you set in code. "parameters" is optional, and you can leave it empty.

You can specify a function that the ChartDataProvider calls when it has received data successfully using "onSuccess".

You have to implement the MVC Controller yourself. The ChartDataProvider expects the MVC Controller to return data as a JSON object.

This is an example of such a JSON object:

```
    {
        "totalRecordCount":0,
        "messages":[
            {
                "id":100,
                "messageType":"info",
                "text":"Contact most likely to buy during next visit."
            }
        ],
        "pageNumber":1,
        "pageSize":5,
        "sorting":[
            {
                "direction":"asc",
                "field":"channel"
            },
            {
                "direction":"asc",
                "field":"device"
            }
        ],
        "filter":[

        ],
        "data":{
            "dataSet":{
                "visits":[
                    {
                        "Channel":10,
                        "IsCampaign":true,
                        "Device":"",
```

```
                        "Location":"Copenhagen, Hovedstaden, Denmark",
                        "Value":"1024",
                        "PageViews":18,
                        "VisitDuration":"01:07:00",
                        "Recency":"2.00:00:00",
                        "StartDateTime":"2013-11-08T22:30:29.2274325Z",
                        "UserAgent":"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36
(KHTML, like Gecko)"
                    }
                ]
            },
            "localization":{
                "fields":[
                    {
                        "field":"channel",
                        "translations":{
                            "40":"RSS",
                            "0":"Unknown",
                            "15":"Search Engine - Organic Branded",
                            "20":"Direct",
                            "50":"Email",
                            "10":"Search Engine - Organic",
                            "90":"Paid",
                            "36":"Referred - Analyst",
                            "31":"Referred - Blog",
                            "34":"Referred - Community",
                            "33":"Referred - Conversations",
                            "32":"Referred - News",
                            "30":"Referred - Other",
                            "35":"Referred - Wiki"
                        }
                    }
                ]
            }
        }
    }
}
```

7.1.1   Properties

Name	Description	Default
Appearance		
Behaviors	The component does not use this property.	
IsVisible	The component does not use this property.	
ToolTip	The component does not use this property.	
Behavior		
AccessKey	The component does not use this property.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID. You do not need to change this property.	Page.Body

Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	The component updates this property with the data it retrieves. This is a read-only property.	
DataURL	Specify the URL of the service the component calls to retrieve data.	
Messages	The component updates this property with the messages the service it calls returns, if any. This is a read-only property.	
QueryParameters	Specify parameters for the service the component calls. This property is optional.	
Design		
ID	Lets you set the ID of the control. You should use a semantic name (for example: NewsFrame).	
Filters		
DateFrom	Specify the first date the component retrieves date from.	
DateTo	Specify the last date the component retrieves date from.	
Resolution	Specify how the component groups data in groups based on time resolution. You can specify Day, Week, Month, Quarter, or Year.	
ResolutionRange	Specify the number of resolution units the component uses to calculate DateFrom and DateTo.	
State		

HasData	The components sets this read-only property True when it has data after getting a response from the service.	
HasNoData	The components sets this read-only property True when it does not have data after getting a response from the service.	
IsBusy	The components sets this read-only property True when it is waiting for a response from the service.	

7.1.2 Configuration

You can create a configuration item for a ChartDataProvider by creating an item based on the ChartDataProvider Parameters template.

The configuration item has these specific fields:

DateFrom
DateTo
Resolution
ResolutionRange

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id
IsBusy
QueryParameters
Data
Messages
HasNoData
DataURL
HasData

7.2 GenericDataProvider

You use the GenericDataProvider to provide data for, for example, a ListControl. You will often use a ListControl for the DetailsPanel of a JourneyChart control.

The GenericDataProvider works the same way as the ChartDataProvider. It also calls an MVC Controller that must return data as a JSON object.

7.2.1 Properties

Name	Description	Default
Appearance		
Behaviors	The component does not use this property.	
IsVisible	The component does not use this property.	
ToolTip	The component does not use this property.	
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
ListSorting	This is a read-only property. You bind to this property from the Sorting property of a ListControl.	
PageNumber	The GenericDataProvider updates this property with the index (number) of the current page when the PageSize is smaller than the total number of pages.	
PageSize	Specify the number of records that the component retrieves from Sitecore.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID. You do not need to change this property.	Page.Body
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on	False

	the data source of the control.	
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	SPEAK updates this property with the data the service returns. This is a read-only property.	
DataURL	Specify the URL of the service the component calls.	
Messages	SPEAK updates this property with the messages the service returns, if any. This is a read-only property.	
QueryParameters	Specify parameters for the service the component calls. This property is optional.	
ServerSorting	Specify how the server sorts data. You specify an array of data sorting directions and data sorting fields. For example: to sort on the field called "visits" and set the sorting direction to "asc", you enter this: {"field": "visits", "direction":"asc"}. You specify "asc" for ascending sort, and "desc" for descending.	
Design		
ID	Lets you set the ID of the control. You should use a semantic name (for example: NewsFrame).	
State		
HasData	The component sets this read-only property True when it has data after getting a response from the service.	
HasMoreData	The component sets this read-only property True when the service has more data than what has already been retrieved.	
HasNoData	The component sets this read-only property True when it does not have data after getting a response from the service.	
IsBusy	The component sets this read-only property True when it is waiting for a response from the service.	

TotalRecordCount	The component sets this read-only property to the number of total records the query retrieved.	

7.3 QueryDataSource

You use the QueryDataSource component to retrieve Sitecore items, using a Sitecore Query or a Sitecore Fast Query.

7.3.1 Properties

Name	Description	Default
Appearance		
Behaviors	The component does not use this property.	
Formatting	Specify a formatting token: • $convert_date_to_friendly_format: convert internal date format to localized format • $send_localized_dates: send localized versions of all dates, so that short and long works as column formatting codes • $convert_id_to_display_name: convert item IDs to the display names of the items You can enter multiple formats as a pipe-separated list.	
Behavior		
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceHolderKey	You do not need to change this property.	Page.Body.
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN.	False
VaryByData	Controls whether output caching varies based on the data source of the component.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on	False

	whether the user has authenticated.	
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Data		
Database	Specify the database that Sitecore queries: • The exact name of a database, for example "core", or "master". • $context_database: context database. • $context_contentdatabase: content database.	The context database
Fields	You can provide a pipe ("|") separated list of field names or IDs that Sitecore retrieves. If you enter an asterix, "*", Sitecore retrieves all fields. Retrieve as few fields as possible to improve performance.	If you do not enter anything, all fields are retrieved.
IsBusy	This read-only property is True when the QueryDataSource is busy retrieving items.	
Items	Contains the retrieved items. You normally bind to this property from a control that can show items, for example the ListControl.	
Language	Specify the language version of the items you want to search in: • Empty: search in all languages. • $context_language: search in current language from the CMS • Exact language: for example: "ja-jp" or "en-us"	All languages
Query	Specify a Sitecore Query or a Sitecore Fast Query.	
Design		
Id	The unique component Id.	
Paging		
PageIndex	The component does not use this property.	
PageSize	The component does not use this property.	

7.3.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

Name	Additional Description
query	
language	
database	
pageSize	
pageIndex	
pageCount	
totalItemsCount	Returns the total number of items found.
items	
fields	
isBusy	

7.3.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
listenTo	
listenToOnce	
once	
set	
stopListening	

7.3.4 Configuration

You can create a configuration item for a QueryDataSource by creating an item based on the QueryDataSource Parameters template.

The configuration item has these specific fields:

Query
IsBusy
Items

It inherits these fields:

Database
Fields
PageSize
PageIndex
Language
Behaviors
Id

7.4 SearchDataSource

You use the SearchDataSource component to retrieve Sitecore items from a specified root item. You can restrict the items that the component retrieves with a search query. You use another control, such as the ListControl, to display the items that the SearchDataSource component retrieves.

By default, the SearchDataSource retrieves all items that match the search query. If you set the PageSize property to a value > 0, the SearchDataSource retrieves the specified number of items at a time. You can tell the SearchDataSource component to retrieve the next "page" of items by raising the "next" event. For example, you can add a button to the page, and set the Click property of this button to invoke the "next" event.

7.4.1 How to configure search

You can specify an initial search by defining a search configuration item and specifying the initial search in Search field of this configuration item. You then reference the configuration item from the SearchConfigItemId property. You use the SearchPanel Config template when you create this item. You can use this item for more than just entering a query string. The fields in the template are described in the following table.

You can create several configuration items that retrieve items in different ways. You can then let users select between these different searches dynamically. One way to do this is:

1. Create a config item based on the SearchPanel Config template for each different search.

2. Configure each of these config items, if necessary.

3. Bind the SearchConfigItemIdm property of the SearchDataSource to the config item that specifies the primary search. Users see the result of this search when initially they load the page.

4. Provide navigation to additional searches. For example, you can create buttons and set the Click property similar to this:

```
set:<name of the data source>({"searchConfig": <item ID of config
item})
```

This is an example:

```
set:DataSource({"searchConfig": "{D502B84F-372E-486B-BDDF-
CE37F54B63DB}"})
```

Remember to create navigation for the "primary search" so that users can return to it after using another search.

The SearchPanel Config template

The template has these fields:

Field	Description
DisplayText	Not used.
Search	Specify the search query string
AppendSearchText	
Root	Point to the item in the content tree that you want the search to start from.
Template	Specify the ID of a template, and the query only retrieves items based on this template.
Base Templates	Specify the names of base templates, and the

	query only retrieves items based on these base templates.
UpdatedWithin7Days	If True, the query only retrieves items updated within the last seven day.
CreatedWithin7Days	If True, the query only retrieves items created within the last seven day.
UpdatedByCurrentUser	If True, the query only retrieves items updated by the current user.
UpdatedBySpecificUser	If you specify the name of a user, the query only retrieves items updated by that user.
CreatedByCurrentUser	If True, the query only retrieves items created by the current user.
CreatedBySpecificUser	If you specify the name of a user, the query only retrieves items created by that user.
Sorting	

Using facets and the FilterControl

You can use facets for search as well. The "Developer's Guide to Item Buckets and Search" describes the concept of facets in more detail. In short, facets are distinct characteristics of items that you can use for grouping. Items can have multiple facets, but they can only have one distinct value for each facet. Examples of facets are Dimensions (size of an image), File Type, or Language.

You use the FilterControl to show facets. Users can select and deselect facets, and the search updates immediately.

You must create the facets you want to present to the users first. Use the Facet template to create an item for each facet and place these facet items in a folder under PageSettings.

The template has these fields:

Field	Description
FieldName	Specify the name of a field that Sitecore uses to create facets. You have to be careful and not specify a field where items can have many different values. If you do that, Sitecore has to create many facets, which degrades performance.
FacetProviderType	

When you have created the facet items, add a FilterControl to the page. Bind the Facets property of this FilterControl to the Facets property of the data source.

Set the FacetsRootItemId of the data source to the item ID of the folder where you have created the facet items, and bind the SelectedFacets property of the data source to the SelectedFacets property of the FilterControl.

7.4.2 Search query syntax

You can search for specific field values with the following syntax:

```
<modifier><field name>:<search string>
```

Where:

- <modifier> defines how Sitecore interprets the search criteria for this field when you enter multiple field names. You can use these modifiers:

- o <no modifier> translates to SHOULD and corresponds to a logical OR

- o + translates to MUST and corresponds to a logical AND

- o – translates to MUST NOT and corresponds to a logical AND NOT

- <field name> specifies the template field name. You must enclose it in quotes if the field name has embedded spaces.

- <search string> is a simple text criteria. You can use * (asterix) as a wildcard. You must enclose the search string in quotes if it has embedded spaces.

You can include search strings for multiple fields, separated by spaces.

The Search field supports sorting using the following syntax:

<field name><sort direction>

Where:

- <field name> specifies the template field name. Enclose in quotes if the field name includes embedded spaces.

- <sort direction> optionally specifies ascending or descending sorting.

  Acceptable values include:

  - o |ASC indicates ascending values.

  - o |DESC indicates descending values.

You can include multiple sort fields.

7.4.3   Properties

Name	Description	Default
Appearance		
Behaviors	The component does not use this property.	
Formatting	Specify a formatting token: • $convert_date_to_friendly_format: convert internal date format to localized format • $send_localized_dates: send localized versions of all dates, so that short and long works as column formatting codes • $convert_id_to_display_name: convert item IDs to the display names of the items You can enter multiple formats as a pipe-separated list.	
Sorting	Specify how the component sorts items. You can bind to another control to make sorting interactive. For example, you often use the ListControl to show the items the SearchDataSource retrieves. You can bind the Sorting property of your SearchDataSource to the Sorting property of the ListControl, and when users click the	

	sorting "arrow" buttons of the ListControl, SPEAK notifies the SearchDataSource.	
Behavior		
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceHolderKey	You do not need to change this property.	Page.Body.
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true.

Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN. | False |
VaryByData	Controls whether output caching varies based on the data source of the component.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Data		
Database	Specify the database that Sitecore searches:

- The exact name of a database, for example "core", or "master".
- $context_database: context database.
- $context_contentdatabase: content database. | The context database |
| Facets | Specify the facets that the SearchDataSource use. | |
| FacetsRootItemId | Points to a configuration item that contains search facets as child items. You create a facet configuration item | |

	based on the Facet template for each facet.	
Fields	You can provide a pipe ("|") separated list of fields that Sitecore should search in. If you enter an asterix, "*", Sitecore searches in all fields. Search in as few fields as possible to improve performance.	If you do not enter anything, all fields are searched.
Items	Contains the retrieved items. You normally bind to this property from a control that can show items, for example the ListControl.	
Language	Specify the language version of the items you want to search in: • Empty: search in all languages. • $context_language: search in current language from the CMS • Exact language: for example: "ja-jp" or "en-us"	All languages
RootItemId	Point to the item in the content tree that you want the search to start from.	The root of the Sitecore content tree.
SearchConfigItemId	Points to a configuration item that contains a search configuration. You create a configuration item based on the SearchPanel Config template in the PageSettings, and use the fields in this item to configure the search.	
SelectedFacets	Specifies which of the available facets the component uses. You normally bind this property to the SelectedFacets property of a FilterControl. When you do this, users can select facets in the FilterControl, and the SearchDataSource updates the search results.	
Text	The text to search for. You normally data bind this value to a text property of another component, for example, a text box where a user can enter search terms. You can use the dropdown list to select properties of the other components that are on the page.	
Design		
Id	The unique component Id.	
Paging		
PageIndex	The SearchDataSource updates this property with the index (number) of the current page when the PageSize is smaller than the total number of pages.	
PageSize	Sets the number of items that the component retrieves from Sitecore.	
State		
HasItems	True if the search finds items (read-only).	
HasMoreItems	True if the search result contains items that the data source has not yet sent to the client (read-only).	

| HasNoItems | True when the data source has sent all items in the search result to the client (read-only). | |
| IsBusy | True if the component is in the process of retrieving data from the server (read-only). | |

7.4.4   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "pageSize" in the following table corresponds to the "PageSize" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
text	
searchConfig	Same as SearchConfigItemId property.
rootItemId	
pageSize	
pageIndex	
totalItemsCount	Returns the total number of items found.
items	
selectedFacets	
facets	
facetsRootItemId	
language	
database	
pagingMode	Specifies how the component handles items from a paged search. Default is "appending", which means it adds the item to the items that are already. This becomes an "infinite scroll".
isBusy	
hasItems	
hasNoItems	
hasMorePages	

7.4.5   View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
next	When you raise this event from PageCode or from a Click property of a control, the SearchDataSource component retrieves the next page of items.
refresh	When you raise this event, the SearchDataSource component refreshes the items it has found by invoking the search again. If you are using paging, this means that the PageIndex is set to 0, and the first page is retrieved.

set	

7.4.6   Configuration

You can create a configuration item for a SearchDataSource by creating an item based on the SearchDataSource Parameters template.

Note
You configure the *search* itself, too. This is described in "How to configure search."

The configuration item has these specific fields:

Formatting
Sorting
Text
FacetsRootItemId
Facets
SearchConfigItemId
Items
RootItemId
SelectedFacets
IsBusy
HasItems
HasNoItems
HasMoreItems

It inherits these fields:

Database
Fields
PageSize
PageIndex
Language
Behaviors
Id

7.5 WebServiceDataSource

You use the WebServiceDataSource to call a web service and receive a response. You specify the following, either as static properties or in code:

- The URL of the web service

- The namespace of the web service

- The method you want the component to call

- Optionally, parameters for the method

The component calls the web service when you invoke the `refresh` method. The response (if any) is available as the `response` model member.

Note the following:

- The component does not change the response in any way.

- If you call a web service in another domain, the remote server must allow cross-domain calls.

7.5.1 Properties

Name	Description	Default
Appearance		
Behaviors	The component does not use this property.	
IsVisible	The component does not use this property.	
Tooltip	The component does not use this property.	
Behavior		
AccessKey	The component does not use this property.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
HasResponse	The component sets this read-only propeety to True when the web service responds. You can retrieve the response from the "response" member.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceHolderKey	You do not need to change this property.	Page.Body
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore	False

---

	caches the HTML generated. For more information regarding caching, please refer to the SDN.	
VaryByData	Controls whether output caching varies based on the data source of the component.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Data		
MethodName	Specify the name of the web service method the component calls.	
Namespace	Specify the namespace of the web service.	
Parameters	Specify the parameters the component passes on when it calls the web service.	
Url	Specify the URL of the web service	
Design		
Id	The unique component Id.	
State		
IsBusy	The component set this read-only property to True when it has called the web service but not yet received a response.	

7.5.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "methodName" in the following table corresponds to the "MethodName" property). The following table has a description when there are exceptions to this convention.

Name	Additional Description
url	
methodName	
nameSpace	
parameters	
requestType	

response	
isBusy	
hasResponse	

7.5.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
listenTo	
listenToOnce	
once	
refresh	
set	
stopListening	

7.5.4 Configuration

You can create a configuration item for a WebServiceDataSource by creating an item based on the WebServiceDataSource Parameters template.

The configuration item has these specific fields:

Url
MethodName
Parameters
Namespace
HasResponse
IsBusy

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

Chapter 8

Dialogs

8.1 DialogWindow

The DialogWindow control shows a dialog that can be modal or not. The content can be internal or external.

You create internal content by adding SPEAK renderings to the placeholders that the DialogWindow control provides.

You specify external content by adding a Frame control to the DialogWindow, and configuring this Frame control to show either a Sitecore dialog or a website.

8.1.1 Opening and closing a DialogWindow

There are three events that you can raise to open or close a DialogWindow:

Event	Description
show	Shows ("opens") the DialogWindow
hide	Hide ("closes") the DialogWindow
toggle	Toogles the current state of the DialogWindow

If you want to show the DialogWindow immediately when a user opens the page that contains the DialogWindow, you can call the "show" event from the PageCode in the containing page.

In many cases, you want to show a button that users can click to open the DialogWindow. You can add this JavaScript to the Click property of the button:

> `show:<ID of DialogWindow>`, for example: show:CustomerLookupWindow

If HeaderLess is False (this is the default), the DialogWindow shows a close button in the header. When a user clicks this "X," the DialogWindow closes.

If EnableKeyboard is True (this is the default), users can close the DialogWindow by pressing Esc.

You can also use the hide event to close the DialogWindow. For example, you can add this to the Click property of a button:

> `hide:<ID of DialogWindow>`, for example: hide:CustomerLookupWindow

The toggle event toggles the state of the DialogWindow: if the DialogWindow is not shown, the toggle event shows it; if it is shown, the toggle event hides it.

8.1.2 Placeholders

The DialogWindow control has four placeholders:

- Actions: place the most important buttons here. If there are many actions associated with the DialogWindow, you can use an ActionControl. If you put controls in this placeholder, the "X" that is normally shown is not visible and clickable. You must make sure that the user has at least one other way of closing the DialogWindow. If you specify a title for the DialogWindow in the Text property, this title is shown in the Actions placeholder.

- Buttons: place secondary buttons here.

- Navigation.

- Content: use this placeholder for the SPEAK renderings you need for the main part of the DialogWindow.

If you have specified that the DialogWindow is HeaderLess, the Action, Buttons, and Navigation placeholders are not shown.

If your DialogWindow is *external,* you normally put a Frame control in the Content placeholder, and you do not use the other placeholders.

8.1.3 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
EnableKeyboard	If True, users can dismiss the dialog with the Esc key. If you set this to False, and you also do not have a header in the dialog, you have to create a way for users to dismiss the dialog yourself.	True
HeaderLess	If True, the dialog does not have a header with a title. In addition, the "X" that users can use to dismiss the dialog is not shown. When you show an external dialog in a Frame, you often want to set HeaderLess to True for aesthetical reasons. If you do, and you want users to be able to close the DialogWindow, you must	False
IsAnimated	If True, the dialog slides in and out when it opens and closes.	False
IsModal	If True, the dialog is modal.	True
IsVisible	If true, the control and the controls it contains are visible.	True
Size	Specify the size of the dialog. You can use these four values: • Small • Medium • Large • X-Large	Medium
Text	Specify a header text for the dialog. The dialog does not show this text if HeaderLess is True.	
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	

Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
NewValue	Specify a number.	
OldValue	Specify a number.	
Design		
Id	Lets you set the ID of the control.	

8.1.4 Configuration

You can create a configuration item for an ArrowIndicator by creating an item based on the ArrowIndicator Parameters template.

The configuration item has these specific fields:

ValueSize
ValueFormat
ImageSize
UpArrowColor
DownArrowColor
EqualSignColor
ValueColor

ShowMinusSign

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

Chapter 9

Forms

9.1 Uploader

An Uploader control lets users select and upload media files. The Uploader control shows an area where users can drag and drop the files from their local computer that they want to upload to the media library. "Drag and drop" does not work in all browsers on all platforms, and local security settings can also disable it. The Uploader offers a more traditional alternative to "drag and drop": a button that users can click to open a file selection dialog.

The Uploader control provides users with a way to select files and a member function to upload selected files. You can raise the "upload" event from a button to trigger this function. Users can then upload selected files by clicking the button. The Uploader uploads *all* currently selected files. The collection of selected files is empty after an upload. The collection of selected files is also empty if the user leaves or refreshes the page.

You can use the UploaderInfo control to show information about the currently selected files (see the separate topic for this control).

9.1.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
BrowseButtonText	Specify the text the browse button shows.	'Browse for media files'
IsVisible	If true, the control is visible.	True
ShortDescriptionText	Specify the text the control shows a label above the Browse button.	'Drag and drop files here'
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true.	False

	Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Database	Specify the database that the Uploader control uploads files to: • The exact name of a database, for example "core", or "master". • $context_database: context database. • $context_contentdatabase: content database.	The context database.
DestinationUrl	Specify the folder where the Uploader puts files that it uploads, for example: /sitecore/media library/Uploader.	/sitecore/media library
Design		
Id	Lets you set the ID of the control.	

9.1.2   Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
destinationUrl	
totalFiles	
globalPercentage	
totalSize	

9.1.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
collection	
fileAdded	
focus	
getTotalSize	
hide	
listenTo	
listenToOnce	
once	
refreshNumberFiles	
set	
show	
stopListening	
toggle	
updateRealImage	
upload	

9.1.4 Configuration

You can create a configuration item for an Uploader control by creating an item based on the Uploader Parameters template.

The configuration item has these specific fields:

Database

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

9.2 UploaderInfo

An UploaderInfo control shows information about files that a user has selected for upload by using an Uploader in the same page. You cannot customize the UploaderInfo control, and you do not have to do any configuration to make it work.

The UploaderInfo control shows the following information about each file:

- Name

- Description

- Alternate Text

- Size (in KB)

- Type (MIME type, for example: image/gif)

The control also shows the button that users can click to remove a file from the set of files that they have selected to upload.

When a user has selected a file for upload, the control shows that the file is "Ready for upload." When the file has been uploaded, the control shows an "Upload completed" message.

Note

The UploaderInfo control shows *all* files the user has selected to upload on the page. If you put two (or more) Uploader controls on one page, the UploaderInfo control shows files from all the Uploader controls. In the UploaderInfo control, you cannot see the Uploader control that each file "belongs to."

9.2.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control is visible.	True
ShowAlternateTextField	If true, the control show the the the alternate text field where the user can enter information about the upload.	True
ShowDescriptionField	If true, the control show the the the description field where the user can enter information about the upload.	True
ShowNameField	If true, the control show the the the name field where the user can enter information about the upload.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	

DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Database	Specify the database that the Uploader control uploads files to: • The exact name of a database, for example "core", or "master". • $context_database: context database. • $context_contentdatabase: content database. The control places the files in the Media Library of the database you specify.	The context database.
Design		
Id	Lets you set the ID of the control.	

9.2.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
files	

9.2.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
addInfo	
collection	
flagFile	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
show	
stopListening	
toggle	
updateInfo	

9.2.4 Configuration

You can create a configuration item for an UploaderInfo control by creating an item based on the UploaderInfo Parameters template.

The configuration item inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

Chapter 10

Indicators

10.1   ArrowIndicator

The ArrowIndicator control shows the difference between two values as an image and a number. The control aligns the image and the number horizontally.

The two values are called OldValue and NewValue. The control shows the following images, depending on the relationship between the two values:

- OldValue > NewValue: the control shows an arrow pointing down
- OldValue < NewValue: the control, shows an arrow pointing up
- OldValue = NewValue: the control shows an equals (=) sign

The number the control shows is the value difference between OldValue and NewValue. You can specify if the control shows this number as the absolute difference or as a percentage. If NewValue if less than OldValue the control automatically shows the number as a negative number.

You can specify the colors and the size the control use for showing the image and the number.

10.1.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
DownArrowColor	Specify the color of an arrow pointing down. You can use all color specifications that are valid in CSS.	Red
EqualSignColor	Specify the color of an equals sign. You can use all color specifications that are valid in CSS.	Gray
ImageSize	Specify the size of the image. You can use these values: Small, Normal, Medium, Large, and ExtraLarge.	Normal
IsVisible	If false, the control does not take up space on the page.	True
ShowMinusSign	Specify if the control shows a minus when the number is negative. You can specify True or False.	False
Tooltip	Specify the text that you want displayed when the user hovers the mouse over	None

	the indicator.	
UpArrowColor	Specify the color of an equals sign. You can use all color specifications that are valid in CSS.	Green
ValueColor	Specify the color of the number value the control shows. You can use all color specifications that are valid in CSS.	Black
ValueFormat	Specify how the control shows the number value. You can specify Number or Percentage.	Number
ValueSize	Specify the size of number value the control shows. You can use these values: Small, Normal, Medium, Large, and ExtraLarge	Normal
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False

VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.DataProvider.Data}	
Design		
Id	Lets you set the ID of the control.	

10.2   ValueLabel

The ValueLabel control shows a value and a label. You can set the size and the color of both.

You can bind the Data property bound to a ChartDataProvider component. The control performs the calculation you specify in DataFunction on the data in the column you specify in DataField.

You can also pass a JSON data set to the control, set a column in this data set, and have the control perform the calculation on this data.

10.2.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If false, the control does not take up space on the page.	True
Label	Specify the text the control displays.	
LabelColor	Specify the color of the text the control shows. You can specify the color in the same formats that are valid in CSS.	Black
LabelPosition	Specify how the control positions the text relative to the number. You can specify either Bottom or Right.	Bottom
LabelSize	Specify the size of the text the control shows. You can specify Small, Normal, Medium, Large, or ExtraLarge.	Normal
Tooltip	Specify the text that you want displayed when the user hovers the mouse over the indicator.	None
ValueColor	Specify the color of the number value the control shows. You can use all color specifications that are valid in CSS.	Black
ValueSize	Specify the size of the number value the control shows. You can use these	Normal

	values: Small, Normal, Medium, Large, and ExtraLarge	
Behaviors		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False

VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Data	Bind to the data the control uses, for example: {Binding.DataProvider.Data} or pass a JSON data set.	
DataMapping		
DataField	Specify the data column the control applies the DataFunction to.	
DataFunction	Specify the function the control applies to the Data column. You can specify Sum, Average, Max, Min, First, or Last.	
Design		
Id	Lets you set the ID of the control.	

Chapter 11

Lists and Grids

11.1   Carousel

A Carousel shows images as a horizontal or a vertical strip. Users can move back and forth between the images in the Carousel by clicking left/right (or up/down) arrows.

To create the images a Carousel shows:

1. Use the Carousel Tile template to create a Carousel item for each image.

2. Enter the Item ID of a Sitecore image you want to show in the Source property of each Carousel Tile item.

By convention, you create the tiles in a folder under PageSettings. You add the tiles to a Carousel by pointing the Tiles property of the Carousel to this folder. The Carousel shows the tiles in the order that they appear in the folder, and it always shows all the tiles.

11.1.1   Properties

Name	Description	Default
Appearance		
Behaviors	Behaviors extend the default behavior of components. You activate a behavior by entering the name, and you can activate multiple behaviors as a space-separated list.	
isVisible	Makes the control visible or invisible. Invisible controls do not take up space in the layout. This means that other controls can change position if a control changes its visibility dynamically.	True.
MinimumTiles	Sets the minimum number of tiles a horizontal Carousel shows, or the exact number of tiles a vertical Carousel shows.	
OnlyEntireTiles	If True, a horizontal Carousel shows only entire tiles.	
Orientation	Horizontal or Vertical.	
TileHeight	Sets the height of the Carousel tiles in pixels.	
TilesPadding	Sets the amount of padding for a tile in pixels.	
TileWidth	Sets the width of the Carousel tiles in pixels.	
ToolTip	Text displayed when the user hovers the mouse over the control.	None.
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute. This is a single character.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	

PlaceholderKey	Text Id of the placeholder where Sitecore renders this control.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this control, with entries varying based on the "VaryBy…" parameters set to true.	False
VaryByData	Controls whether output caching varies based on the data source of the control.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Data		
Tiles	ID of folder that has tile items.	
Design		
Id	Lets you set the ID of the control.	

11.2   ListControl

A ListControl control displays items. It does not contain any data itself, and it does not retrieve any items. You must use data binding to provide the items that the ListControl shows. Typically, you bind a ListControl to a data source component, such as the SearchDataSource, in the Items property of the ListControl.

A ListControl represents items as icons (when you set ViewMode to Icon) or text in rows and columns (when you set ViewMode to DetailList). The DataSource property points to a configuration item that specifies which fields to show in the columns for the DetailList view mode.

The ListControl shows all items that the data source provides. Data source components, such as the SearchDataSource, provide support for server-side pagination and infinite scrolling.

11.2.1   Properties

Name	Description	Default
Appearance		
AllowMultipleColumnSorting	If false, users can only sort the ListControl columns one at a time. When the ListControl sorts a column, it will set the other columns to the "unsorted" state.	True
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
Height	Specifies the height of the control in pixels.	None.
isVisible	Makes the list control visible or invisible. Invisible list controls do not take up space in the layout. This means that other controls can change position if a control changes its visibility dynamically.	True.
Sorting	You use this property this way: • You can bind a data source that can sort data (such as the SearchDataSource) to ListControl.Sorting. When a user changes the sorting direction, the ListControl will notify the data source. • You can read and write the value of Sorting from JavaScript. The ListControl usrs this convention for the three distinct sorting values: "a"+fieldname: ascending sort "d"+fieldname: descending sort empty: no sort direction If, for example, you have a field called "itemName", it will have these sorting value: "aitemName", "ditemName", and the empty string.	

	You can set the sorting direction like this: `javascript:app.ListControl1.set("sorting","aitemName");` If you specify more than one sortable field, the sorting value is a pipe-separated list of the sorting values.	
ToolTip	Specify the text that you want displayed when the user hovers the mouse over the button. If your dialog or application supports multiple languages, do not specify a static text in the component properties. See the DataSource property for an alternative way to specify text	None.
ViewMode	Specifies how the control shows items: • IconList shows icons • DetailList: shows the columns that you have set up in a configuration item (see the description of the DataSource property). In IconList mode, ListControl shows Thumbnails for items when the item has such a field, and Icons if items do not have Thumbnails.	DetailList.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create a configuration item based on the ListControl template in the PageSettings folder, and configuration items based on the ColumnField emplate as children of this item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False

VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
SelectedItemId	Sets or gets the ID of the currently selected item. You do not normally set the value, as is the control updates the value when the user makes a selection.	
Data Bindings		
Items	Points to the items the control shows. You can use the dropdown to select a binding to another control. You typically select a binding to the items property of a data source component.	
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: HousePhotoList).	

11.2.2   Model Members

Name	Description
isVisible	
items	
selectedItem	
selectedItemId	
sorting	
empty	

11.2.3   View Members

Name	Description
addBehavior	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
setDetailListView	
setIconView	

show	
stopListening	
toggle	

11.2.4 Configuration

You can create a configuration item for a ListControl by creating an item based on the ListControl Parameters template.

The configuration item has these specific fields:

Value
ViewMode
Height
Sorting
Items
SelectedItemId

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

11.2.5 The ListControl and the ColumnField templates

You specify some of the functionality of a ListControl in configuration item. This configuration is necessary if you use the DetailList view mode.

Specify the configurations like this:

1. Add an item based on the ListControl as a child of the your PageSettings.

2. Add an item based on the ColumnField template as a child of the ListControl item – one for each column of data you want to show.

You do not have to edit anything in the ListControl item.

The ColumnField template has these fields:

Field	Description
HeaderText	Specify a column heading. The control uses this in DetailList mode.
DataField	Specify the name of a data field for this column. Enter the name exactly as it is in the template of the items you want the ListControl to show. You can also use one of the following tokens instead of a data field: itemId (the Item ID) itemName (the Item Name) $displayName (the Display Name) $database (name of the database the item belongs to) $language (the language of the item) $version (the version of the item) $templateName (the name of the template the item is based on) $templateId(the item ID of this template)

	$hasChildren (shows whether the item has child items or not) $path (the Sitecore path of the item) $url (the Sitecoore URL of the item) $mediaurl (the Sitecore URL of the icon asset ofm the item) $icon (the Sitecpre path of the icon of the item)
Sortable	If you select this field, the ListControl is sortable and it displays a sorting button that users can click. This button has three states that it will cycle though when a user clicks it: no direction, ascending, and descending. The ListControl does not do the actual sorting: a data source component (such as the SearchDataSource) can bind to Binding.ListControl.Sorting to listen to the sorting direction. It is up to the data source component to use this information.
SortField	Specify the name of the index field if it is different from the content field name. If the fields do not have the same name, sorting will not work unless you specify SortField.
Formatter	Specify formatting strings. You can specify formatting of dates, numbers, and percentages. **Dates**: You can use yy, yyyy, m, mm, d, and dd to format date values. For example: "m-d-y" formats a date as "11-23-14", and "dd/mm/yyyy" as "02/09/2011". yy and yyyy: the year as either two digits or four digits. m and mm: the month without or with a leading zero when needed. d and dd: the day without or with a leading zero when needed. h and hh: hour without or with a leading zero when needed. m and mm: minutes without or with a leading zero when needed. You can also specify date formatting in the SearchDataSource.If you specify formatting in the data source, it will overrule formatting you specify in a ColumnField Formatter field. **Numbers and percentages**: You can specify how the ListControl formats number and percentages. The formatting uses the rules of the context language for thousands and decimal separators. Specify number formatting like this: "#Nx", where "x" is the number of decimals. If x is absent, the ListControl uses a default of 2. Specify percentage formatting like this: "#Px", where "x" is the number of decimals. The ListControl will append a percentage sign ("%") to the number and format the number just as it does for the "N" formatting. It does convert a number to a percentage.
HTML Template	See "Using an HTML Template".
HeaderAlignment	Specify the alignment of the header text of the column. You can use Center, Left, and Right.
ContentAlignment	Specify the alignment of the content of the column. You can use Center, Left, and Right.
Width	Specify the width of the column in pixels or in percent of the width of the complete table.

11.2.6 Using an HTML Template

You can format the DetailList view with an HTML template. In this way, you can control the formatting yourself and you can create hyperlinks, for example.

You specify an HTML template this way:

Add a configuration item based on the ColumnField template for each column you want in the table that the ListControl creates.

Specify HeaderText but not DataField in these items. This is the column heading.

Specify a formatting string in the HTML Template field. You can insert fields from the retrieved items like this: `{{<fieldname}}` where `fieldname` is the exact name of the field in the template that the retrieved item is based on. The name is case-sensitive.

Note
You can freely use *any* fields from the retrieved items in each HTML Template.

If the items that the ListControl show do not all have the same fields, it is possible that some items do not have a field you specified in the HTML template. In this case, the ListControl outputs the field placeholder as is (for example: `{{Dimensions}}`).

An example:

If the retrieved items have fields called "url", "_Created", "Alt", "image", and "itemName", the following HTML Template creates a hyperlink with the image and the itemName as the visible part, and the url as destination:

```
<a href="{{url}} title="Item was created{{_Created}}"><img alt="{{Alt}}"
src="{{image}} /> {{itemName}}</a>
```

11.3 TreeView

A TreeView shows items in a tree structure, starting from the item that you define in the RootItem property.

Users can fold and unfold the branches of the tree, select individual items, or select ("checkbox") multiple items.

You can specify that the TreeView control displays a check box for each item. Users can only "check" items if the TreeView control displays checkboxes.

Selecting an item and "checking" an item are not the same in a TreeView control. Items can be selected and not checked, or they can be checked and not selected. The CheckedItemIds property is a list of the IDs of all checked items, and the SelectedItemId is the ID of the currently selected item.

The branches of the tree are the folders of the path from the RootItem to the items, and the items are the leaves. The TreeView shows branches and leaves as text and an icon. It uses the name of the folder or item as the text, and you can decide whether it uses the Sitecore icons or generic icons.

11.3.1 Properties

Name	Description	Default
Appearance		
Behaviors	Behaviors extend the default behavior of components. You activate a behavior by entering the name, and you can activate multiple behaviors as a space-separated list.	
isVisible	Makes the list control visible or invisible. Invisible list controls do not take up space in the layout. This means that other controls can change position if a control changes its visibility dynamically.	True.
ToolTip	Text displayed when the user hovers the mouse over the control.	None.
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute. This is a single character.	
ClickFolderMode	Specifies behaviour on folder click. Can be 1:activate, 2:expand, or 3:activate and expand	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item. Use a configuration item based on the TreeView Parameters template, and point to this item here.	
IsCheckboxEnabled	If True, TreeView shows a check box for each node in the tree. Users can check and uncheck nodes by clicking. If you click a node that has children, TreeView checks or unchecks this node as well as all child nodes.	True

	Note A "checked item" is not the same as a "selected item." TreeView updates the CheckedItemsIds property with a list of currently checked items when an item is checked or unchecked.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Text ID of the placeholder where Sitecore renders this control.	
SelectMode	Specify how the user selects nodes in the tree: "1": only one node can be selected at a time. Selecting a node deselects any node that was already selected. "2": multiple nodes can be selected. "3": multi-hierarchal: child nodes are selected when a node is selected.	3 (multi-hierarchal)
ShowHiddenItems	If true, the TreeView will show items where the _Hidden property is True. If you change the setting of this property at runtime (from JavaScript code), the TreeView control will not use the new setting for nodes in the tree that are already expanded.	False
ShowSitecoreIcons	If True, TreeView shows item icons; if False, TreeView shows generic icons (one for folders, another for items).	False
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this control, with entries varying based on the "VaryBy…" parameters set to true.	False
VaryByData	Controls whether output caching varies based on the data source of the control.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Data		
CheckedItemIds	A pipe ('|') separated read-only list of the Item IDs of checked items. TreeView updates the list when an item is checked or unchecked. This property is read-only.	

PreLoadPath	Specify a path to an item that is selected when the control is rendered. You must specify the path like this: <Item ID > / <Item ID> / … The first item in this path must be the RootItem. If the path has an error, the TreeView is still shown, but nothing is selected.	
RootItem	The ID of the Sitecore item TreeView starts from.	The top of the context database content tree.
SelectedItemId	The Item ID of the currently selected item. This property is read-only.	
Design		
Id	Lets you set the ID of the control.	

11.3.2   Model Members

Name	Description
isVisible	
selectedItemId	
selectedItemPath	
checkedItemIds	
pathToLoad	
isBusy	
selectedNode	
showSitecoreIcons	
isCheckboxEnabled	
isKeyboardSupported	
isPersist	
isAutoFocus	
isAutoCollapse	
clickFolderMode	
selectMode	
isNoLink	
debugLevel	

11.3.3   View Members

Name	Description
addBehavior	
appendLoadedChildren	
disable	
enable	
focus	
getActiveNode	
getRoot	
getSelectedNodes	

getTree	
hide	
initialized	
listenTo	
listenToOnce	
loadKeyPath	
nodeExpanding	
onActivate	
onSelect	
once	
raiseEvent	
set	
show	
stopListening	
toggle	

11.3.4   Configuration

You can create a configuration item for a TreeView by creating an item based on the TreeView Parameters template.

The configuration item has these specific fields:

RootItem
PreLoadPath
SelectedItemId
CheckedItemIds
ClickFolderMode
IsCheckboxEnabled
SelectMode
ShowSitecoreIcons

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

Chapter 12

Navigation

12.1 ApplicationLauncher

An ApplicationLauncher control shows "application shortcuts." These shortcuts open a SPEAK application when users click them. You can use the ApplicationLauncher to create a dialog that serves as a "launch pad" for one or more SPEAK applications.

12.1.1 How to create shortcuts

You add shortcuts to the ApplicationLauncher in this way:

1. Create a folder for your shortcuts as a subfolder of the PageSettings folder

2. Create a shortcut item for each shortcut you need. You create a shortcut item using the Application Shortcut template branch.

   Now, you must make the following changes to each of your shortcut items:

3. Change the Tooltip field to something meaningful.

4. Select an icon for the shortcut.

5. Specify the Item ID of the page that SPEAK opens when users click the shortcut. You do this in the Starting Page ID field (the "main page" of your application).

12.1.2 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control and the controls it contains are visible.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Point to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this	False

| | component, with entries varying based on the following "VaryBy…" parameters set to true.

Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN. | |
|---|---|---|
| VaryByData | Lets you specify whether output caching varies based on the data source of the control. | False |
| VaryByDevice | Lets you specify whether caching varies based on the name of the context device. | False |
| VaryByLogin | Lets you specify whether output caching varies based on whether the user has authenticated. | False |
| VaryByParameters | Lets you specify whether output caching varies based on rendering parameters passed to the presentation component. | False |
| VaryByQueryString | Lets you specify whether output caching varies based on query string parameters passed in the URL. | False |
| VaryByUser | Lets you specify whether output caching varies by the domain and username of the context user. | False |
| **Data** | | |
| RootItemId | Point to a folder that contains shortcut items. | |
| **Design** | | |
| Id | Lets you set the ID of the control. You should use a semantic name (for example: CloseButton). | |

12.1.3   Configuration

You can create a configuration item for an ApplicationLauncher control by creating an item based on the ApplicationLauncher Parameters template.

The configuration item has this field:

RootItemId

12.2   Breadcrumb

A Breadcrumb control shows a "breadcrumb trail" showing the parent pages of the current page. Each page in the hierarchy formed is shown as a link to the page; the links are separated by ">" characters. The Breadcrumb control shows the top page link in the hierarchy as an icon, and it shows all other page links as text links, where the text is the name of the page.

You create the hierarchy by setting up a hierarchy of Breadcrumb items. You create a Breadcrumb item using the Breadcrumb template, and you set the Page field of each Breadcrumb item to the ID of the page you want at that level of the hierarchy.

You add the Breadcrumb control to all pages that are part of the hierarchy. You set the RootItemId of the Breadcrumb control in all of the pages to the same value: the ID of the page that is at the top of the hierarchy.

The Breadcrumb shows the list of links (the "breadcrumb trail") down to and including the current page, always starting from the page at the top of the hierarchy.

12.2.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control and the controls it contains are visible.	True
ToolTip	The Border control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	

Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
RootItemId	Specify the ID of the Breadcrumb item that is at the top of the page hierarchy you have created with Breadcrumb items.	
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

12.2.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
width	
height	

12.2.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
addBehavior	
focus	
hide	
listenTo	
listenToOnce	
once	
set	
show	
stopListening	
toggle	

12.2.4 Configuration

You can create a configuration item for a Breadcrumb control by creating an item based on the Breadcrumb Parameters template.

The configuration item has these specific fields:

RootItemId

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

12.3 HyperlinkButtonsGroup

A HyperlinkButtonGroup control lets you display a group of HyperlinkButton controls together. You define the set of HyperlinkButtons using configuration items, either in PageSettings for an individual page, or as shared settings for a number of pages.

You can combine a HyperlinkButtonsGroup with a set of HyperlinkButton controls, for example, to create a "links menu" that works consistently across a set of pages when you reuse the HyperlinkButtons configuration items from shared settings.

12.3.1 Properties

Name	Description	Default
Appearance		
Behaviors	The component does not use this property.	
IsVisible	If false, the group and its child buttons take up no space on the page.	
ToolTip	Text displayed when the user hovers the mouse over the HyperlinkButtonsGroup.	None.
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute. This is a single character.	
DataSource	Points to a configuration item folder. By convention, you create this as a subfolder under the PageSettings folder under the page definition item; or as a subfolder of a shared settings folder of your choice. If you create items in this folder and use the HyperlinkButton template, these items are shown in the HyperlinksButtonsGroup container.	
Parameters		
PlaceholderKey	Text Id of the placeholder where Sitecore renders this control.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN.	False
VaryByData	Controls whether output caching varies based on the data source of the control.	False

VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Design		
Id	Lets you set the ID of the control.	

12.3.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table has a description when there are exceptions to this convention.

Name	**Additional Description**
isVisible	
selectedItemName	
selectedItemId	

12.3.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	**Description**
_selectItem	
focus	
hide	
preventIfDisable	
select	
set	
show	
toggle	

12.3.4 Configuration

You can create a configuration item for a HyperlinkButtonsGroup by creating an item based on the HyperlinkButtonsGroup Parameters template.

The configuration item has these specific fields:

IsEnabled
SelectedItemId

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

Chapter 13

Resources

13.1 Rule

The Rule component triggers a rule when another SPEAK component raises an event. You specify the component and the event that make the Rule component trigger a rule.

A rule consists of a *condition* and an *action*. You create a rule by creating a Rule item based on the RuleDefinition template. You connect the Rule component to a Rule item in the RuleItemId property of the Rule component.

You add a Rule component to your dialog for each event that you are interested in, and a dialog can have multiple Rule components. Several different Rule components can point to the same Rule item.

SPEAK executes the rules of a dialog *after* the PageCode for the dialog has finished executing.

13.1.1 How to configure the Rule component

You specify the event that triggers a rule like this:

1. Specify the ID of the SPEAK component that you want to "listen" to in the TargetControl property.

2. Specify the event in the Trigger property. You can specify just the event (for example: "change", or "click"), or you can specify an event and an attribute (for example: "change:facets"). If, for example, you specify "change:facets", the Rule component triggers a rule when the component raises the "change" event for the "facets" attribute. The attributes are specific for each SPEAK component, and you can find the attributes for a component in the list of Model Members in the reference page for the component.

The default component is "window." This means that if you do not specify a component, the Rule component listen for events raised by the "window" component. The "window" component is the dialog page itself. It has a "resize" event.

Components can have specific events, but all components have a "change" event. You can use a combination of the "change" event, an attribute, and the Rule condition (as described in the following section) to precisely specify when a Rule is executed.

13.1.2 How to create a rule

1. First, create a Rule item, using the RuleDefinition template.

2. You add rules to the Rule item in the Rule field. Click the Edit button next to the field to open the Rule Editor.

3. The Rule Editor has three panes: Conditions and Actions where you can select conditions and actions, and a pane where you see and edit the rules (the Rule pane).

4. The Rule Editor creates an empty rule if there are no rules when you open it. If there are rules, you can add conditions and actions to an existing rule or you can create a new rule. You right-click and select Add New Rule from the context-menu to create a new rule.

5. You add conditions and actions to a rule by double-clicking in the Conditions or in the Actions pane. This inserts the condition or the action in the Rule pane. If a condition or an action is already there, the new one is added. You can use the context-menu to remove conditions and actions from the Rule pane.

13.1.3   Conditions

When SPEAK executes a rule, the first thing that happens is that the conditions are evaluated. A condition looks this:

"where component name has a selected item"

After you have inserted a condition like the one shown in the Rule pane, you have to edit it. The editor shows placeholders in blue. In this condition, "where" and "name" are placeholders. You specify a value for a placeholder by double-clicking.

The "where" placeholder can only have two values: "where" and "except where." When you click the "where" placeholder, the value changes back and forth between these two values.

The "name" placeholder is, as you can see from the context, the name (ID) of a SPEAK component. Click the placeholder, and enter the ID of the component.

If you add another condition, it is inserted with an "and" placeholder. You can click this placeholder to change it to "or."

The conditions of a rule can evaluate to either true or false. If the conditions of a rule evaluate to true, SPEAK executes the actions of the rule. If the conditions evaluate to false, nothing else happens for that rule.

Note
You can specify conditions for other components than the component that raised the event that triggered the rule.

13.1.4   Actions

You add actions to the Rule pane in a similar way to adding conditions. Actions contain placeholders as well.

An action could be:

"remove errors from the MessageBar name", where "name" is a placeholder. You edit this action, and enter the ID of the MessageBar in the placeholder.

You can add multiple actions to one rule. Actions are always "and'ed".

You can always see the full current set of conditions and actions in the two panes in the Rule Editor. You can also see the actions and conditions in the Content Editor under Speak/Layouts/Renderings/Resources/Rule/Rules.

13.1.5   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control and the controls it contains are visible.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.

Behavior		
DataSource	Point to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	You do not have to change this.	Page.Body
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Field	Specify the name of the field in the rule item that contains the rule configuration. SPEAK sets the value to "Rule", and you should not change it.	Rule
RuleItemId	Point to the rule item that contains the conditions and the actions of this rule.	
TargetControl	Specify the ID of the SPEAK control that the Rule component listens to for the event specified in Trigger.	The Rule component listens to the "window" for events if you do not specify a control.
Trigger	Specify the event that triggers the rule. You can specify an event ("change"), or an event and an attribute ("change:text").	
Design		
Id	Lets you set the ID of the control.	

13.1.6 Model Members

Name	Additional Description

13.1.7 View Members

Name	Description
addBehavior	
evaluate	
listenTo	
listenToOnce	
once	
set	
stopListening	
subscribe	

13.1.8 Configuration

You can create a configuration item for a Rule component by creating an item based on the Rule Parameters template.

The configuration item has these specific fields:

Trigger
RuleItemId
Field
TargetControl

It inherits these fields:

Behaviors
Id

13.2 SettingsDictionary

You use the SettingsDictionary component to retrieve strings from the web.config file, and make these string available to client-side JavaScript code.

When you have created and configured a SettingsDictionary component, you can use in JavaScript in your page. For example:

```
javascript:alert(SettingsDictionaryId.get("settingName"))
```

13.2.1 How to Configure a SettingsDictionary Component

You configure a SettingsDictionary component in this way:

1. Create a configuration item based on the SettingsDictionary Parameters template.

2. Specify the settings you want retrieved in the Settings field. You can add multiple settings by separating them with commas.

3. Point the DataSource property of the SettingsDictionary component to the configuration item.

13.3   StringDictionary

You use the StringDictionary component to retrieve strings from the Sitecore dictionary.

The recommended way to add translatable text strings to a SPEAK application is to use configuration items. Configuration items are regular Sitecore items, and there are procedures for translating text fields. However, you cannot use the configuration items from client-side JavaScript code. You can use a StringDictionary component instead because this component makes dictionary strings available for client-side code.

You can add strings from the dictionary to the StringDictionary component in two ways:

- You can point to one or more dictionary items. The String Dictionary component makes these dictionary items available.

- You can specify one or more text strings (*phrases.*) The StringDictionary component finds these phrases in the dictionary and makes the corresponding dictionary items available.

When you have created and configured a StringDictionary component, you can use it in JavaScript in your page. For example:

```
Sitecore.Resources.Dictionary["somePhrase"]
```

or

```
Sitecore.Resources.Dictionary.translate("somePhrase")
```

"translate" returns the key ("somePhrase") if it cannot find a translation. If you do not use "translate", you will get "undefined" in this case.

13.3.1   How to Configure a StringDictionary Component

You configure a StringDictionary component in this way:

1. Create a configuration item based on the StringDictionary Parameters template.

2. You point to items in the Sitecore dictionary like this:

    a. Open the configuration item in the Content Editor (Sitecore Rocks does not work for this for the time being.)

    b. Edit the Dictionary field in the configuration item. You can select as many dictionary items you need.

    c. Save the configuration item.

3. Alternatively, you specify *phrases* in the Phrases field. You can add multiple phrases by separating them with pipes ('|').

4. Point the DataSource property of the StringDictionary component to the configuration item.

Chapter 14

Searching and Filtering

14.1 FilterControl

The FilterControl control shows the facets that you have configured for a data source. Users can select and deselect facets, and the FilterControl control tells the data source what the currently selected facets are each time a user changes the selection.

14.1.1 Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control and the controls it contains are visible.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False

VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Data		
Facets	Specify the facets users can select by binding to the Facets property of the data source.	
SelectedFacets	Contains the currently selected facets. You normally bind the SelectedFacets property of the data source to the property here.	
Design		
Id	Lets you set the ID of the control. You should use a semantic name (for example: CloseButton).	

14.1.2 Model Members

Model members represent the properties of a component. You can manipulate the model members in the PageCode.

The model members correspond to the parameters with the same name described previously (note that case is not important in this situation, for example, "isVisible" in the following table corresponds to the "IsVisible" property). The following table displays a description when there are exceptions to this naming convention.

Name	Additional Description
isVisible	
items	
isLoading	

14.1.3 View Members

View members represent events that the component either raises and/or listens to. You can listen for and raise events in the PageCode.

Name	Description
add	
addBehavior	
focus	
hide	
listenTo	
listenToOnce	
once	
renderItem	
renderItems	
reset	
set	
show	
stopListening	
toggle	

14.1.4   Configuration

You can create a configuration item for a FilterControl by creating an item based on the FilterControl Parameters template.

The configuration item has these specific fields:

Items

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

14.2 SearchPanel

A SearchPanel control provides three placeholders:

- Filters

- Searches

- Details

You use the control and the placeholders when you design a SPEAK dialog that lets users search for items. When you use the SearchPanel control for the different elements of a search page, the SearchPanel aligns and positions these elements correctly.

Use the placeholders like this:

Filters
If you use a FilterControl in your search page, you often also have a ToggleButton that users can click to show and hide the content of the filter control. If you use such a button, put it in this placeholder.

Searches
You usually have a text box for user input and an IconButton for starting a search in a search page. When you do, place them in this placeholder.

Details
You use this placeholder for the component that displays the search results. If you use a FilterControl it also has to be in this placeholder.

14.2.1 Properties

Name	Description	Default
Appearance		
Behaviors	Behaviors extend the default behavior of components. You activate a behavior by entering the name, and you can activate multiple behaviors as a space-separated list	
IsVisible	If true, the control and the control it contains are visible.	True
ToolTip	Text displayed when the user hovers the mouse over the control.	None.
Behavior		
AccessKey	Specifies the HTML5 AccessKey attribute. This is a single character.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceholderKey	Text Id of the placeholder where Sitecore renders this control.	
Caching		

Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true.	False
	Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN.	
VaryByData	Controls whether output caching varies based on the data source of the control.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Design		
Id	The control Id.	

14.2.2   Configuration

You can create a configuration item for a SearchPanel control by creating an item based on the SearchPanel Parameters template.

The configuration item inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

Chapter 15

Services

15.1 EntityDataSource

You use the EntityDataSource to fetch one entity ("business object") from a repository. The component uses the EntityService from Sitecore.Services.Client to fetch the entity. See the "Sitecore.Services.Client: Concept, Reference and How-Tos" document for information about the EntityService.

15.1.1 Properties

Name	Description	Default
Appearance		
Behaviors	The component does not use this property.	
IsVisible	The component does not use this property.	
Tooltip	The component does not use this property.	
Behavior		
AccessKey	The component does not use this property.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceHolderKey	You do not need to change this property.	Page.Body
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN.	False
VaryByData	Controls whether output caching varies based on the data source of the component.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain	False

	and username of the context user.	
Data		
EntityID	The unique ID of the entity you want to fetch.	
IsBusy	True if the component is in the process of retrieving data from the server (read-only).	
ServiceUrl	The root end point of the RESTful API of the service, for example: /api/Sitecore/ssc/product.	
Design		
Id	The unique component Id.	

15.1.2   Configuration

You can create a configuration item for an EntityDataSource by creating an item based on the EntityDataSource Parameters template.

The configuration item has these specific fields:

ServiceUrl
EntityID
IsBusy

It inherits these fields:

IsVisible
AccessKey
Tooltip
Behaviors
Id

15.2   StoredQueryDataSource

You use the StoredQueryDataSource to fetch items from Sitecore. The component uses the ItemService from Sitecore.Services.Client to fetch theitems. See the "Sitecore.Services.Client: Concept, Reference and How-Tos" document for information about the ItemService.

15.2.1   Properties

Name	Description	Default
Appearance		
Behaviors	The component does not use this property.	
Behavior		
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name value pairs. An alternative method for assigning property values.	
PlaceHolderKey	You do not need to change this property.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the HTML generated. For more information regarding caching, please refer to the SDN.	False
VaryByData	Controls whether output caching varies based on the data source of the component.	False
VaryByDevice	Controls whether caching varies based on the name of the context device.	False
VaryByLogin	Controls whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Controls whether output caching varies based on rendering parameters passed to the presentation component.	False
VaryByQueryString	Controls whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Controls whether output caching varies by the domain and username of the context user.	False
Data		

Database	Specify the database to retrieve the item from.	
Fields	Specify the names of the fields to retrieve in a comma-separated list.	
IncludeStandardTemplateFields	If true, the standard template fields are part of the data that is retrieved.	
IsBusy	True if the component is in the process of retrieving data from the server (read-only).	
Items	This read-only property ontains the retrieved items.	
Language	Specify the language version of the items you want to search in.	
QueryID	Specify the id of the Sitecore item that contains the definition of the query to run.	
Design		
Id	The unique component Id.	
Paging		
PageIndex	Specify the page number in the result set of pages that the component shows.	
PageSize	Specify the number of results the component returns.	

15.2.2   Configuration

You can create a configuration item for a StoredQueryDataSource by creating an item based on the StoredQueryDataSource Parameters template.

The configuration item has these specific fields:

QueryID
IsBusy
Items
IncludeStandardTemplateField

It inherits these fields:

Database
Fields
PageSize
PageIndex
Language
Behaviors
Id

Chapter 16

Silverlight

16.1   SilverlightFrame

You use the SilverlightFrame control to embed a Silverlight plug-in in a SPEAK page. You can set a number of properties, and SPEAK uses these properties to instantiate the Silverlight plug-on. You cannot interact with the Silverlight plug-in in JavaScript at runtime from SPEAK.

16.1.1   Properties

Name	Description	Default
Appearance		
Behaviors	A Behavior extends the default functionality of a component. You activate a behavior by entering the name of the behavior. You can enter several behaviors as a space-separated list.	
IsVisible	If true, the control is visible.	True
ToolTip	The control displays this text when users hover the mouse over the control.	None.
Behavior		
AccessKey	Lets you specify a single character that SPEAK renders as the HTML5 AccessKey attribute of the control.	
DataSource	Points to a configuration item. By convention, you create the configuration item in the PageSettings folder under the page definition item.	
Parameters	Text string with name-value pairs. An alternative method for assigning property values.	
PlaceholderKey	Sitecore renders the control in the placeholder that has this ID.	
Caching		
Cacheable	If true, Sitecore caches the HTML generated by this component, with entries varying based on the following "VaryBy…" parameters set to true. Sitecore generates SPEAK component HTML on the server side and these settings influence how Sitecore caches the generated HTML. For more information on caching, refer to the SDN.	False
VaryByData	Lets you specify whether output caching varies based on the data source of the control.	False
VaryByDevice	Lets you specify whether caching varies based on the name of the context device.	False
VaryByLogin	Lets you specify whether output caching varies based on whether the user has authenticated.	False
VaryByParameters	Lets you specify whether output caching varies based on rendering parameters passed to the presentation component.	False

VaryByQueryString	Lets you specify whether output caching varies based on query string parameters passed in the URL.	False
VaryByUser	Lets you specify whether output caching varies by the domain and username of the context user.	False
Code		
ParseParameterType		
Data		
AppName	Specify the name of the Silverlight plug-in.	
AutoUpgrade	If true, a Silverlight plug-in that is an earlier version than the version specified in MinRuntimeVersion will attempt to update automatically.	
Height	Specify the height of the frame in pixels or percent.	100%
InitParams	Specify a string that represents a set of user-defined initialization parameters. The format of the value is also user-defined.	
MinRuntimeVersion	Specify the version of Silverlight that is required by the Silverlight component. See also then AutoUpgrade property.	The currently installed version, or null if Silverlight is not installed.
OnError	Specify the name of the function that Silverlight calls when the plug-in generates a parse or run-time error at the native-code level.	Null
OnLoad	Specify a handler for the Loaded event that occurs when the Silverlight plug-in has finished loading into the browser DOM.	Null
ScriptFile	Specify the name of a file that contains JavaScript for the Silverlight component. Note: this is JavaScript the component uses, and not JavaScript that SPEAK executes.	
Source	Specify the URI of the content or package to load into the Silverlight plug-in.	Null
Width	Specify the width of the SilverlightFrame control in pixels or percent.	100%
WindowLess	If True, the Silverlight plug-in is displayed as a windowless plug-in.	True
Design		
Id	Lets you set the ID of the control.	
SilverlightObjectId	SPEAK sets the value of this property to the ID that the Silverlight component has in the DOM of the page. You cannot change this property at run-time.	

Chapter 17

Structures

17.1 Substructures

17.1.1 ApplicationContentM

An ApplicationContentM control is a page substructure control. You use it as a substructure for any of the Dashboard, List, or Task page structure controls. It provides this placeholder:

- ApplicationContent.Main – for the main content

You must put an ApplicationContentM control in the ApplicationContent placeholder that the page structure control provides.

17.1.2 ApplicationContentMI

An ApplicationContentMI control is a page substructure control. You use it as a substructure for any of the Dashboard, List, or Task page structure controls. It provides this placeholder:

- ApplicationContent.Main – for the main content

- ApplicationContent.Info – for extra information

You must put an ApplicationContentMI control in the ApplicationContent placeholder that the page structure control provides.

17.1.3 ApplicationContentNM

An ApplicationContentNM control is a page substructure control. You use it as a substructure for any of the Dashboard, List, or Task page structure controls. It provides these placeholders:

- ApplicationContent.Navigation – for navigation (for example, a HyperLinkButtonsGroup)

- ApplicationContent.Main – for the main content

You must put an ApplicationContentNM control in the ApplicationContent placeholder that the page structure control provides.

17.1.4 ApplicationContentNMI

An ApplicationContentNMI control is a page substructure control. You use it as a substructure for any of the Dashboard, List, or Task page structure controls. It provides these placeholders:

- ApplicationContent.Navigation – for navigation (for example, a HyperLinkButtonsGroup)

- ApplicationContent.Main – for the main content

- ApplicationContent.Info – for more information

You must put an ApplicationContentNMI control in the ApplicationContent placeholder that the page structure control provides.

17.1.5 ApplicationHeader

An ApplicationHeader control is a page substructure control. You use it as a substructure for any of the Dashboard, List, or Task page structure controls. It provides these placeholders:

- ApplicationHeader.BreadCrumb – for a BreadCrumb control

- ApplicationHeader.ContextSwitch –

- ApplicationHeader.Title – for the page title

- ApplicationHeader.Actions – for an ActionControl

You must put an ApplicationHeader control in the ApplicationHeader placeholder that the page structure control provides.

17.1.6   DialogContentMI

A DialogContentMI control is a page substructure control. You use it as a substructure for a Dialog page structure control. It provides three placeholders:

- DialogContent.CommandsBar – for buttons and other controls that let users execute commands

- DialogContent.Main – for the main content

- DialogContent.Info – for additional information

You must put a DialogContentMI control in the DialogContent placeholder that the Dialog control provides.

17.1.7   DialogContentNM

A DialogContentNM control is a page substructure control. You use it as a substructure for a Dialog page structure control. It provides three placeholders:

- DialogContent.Navigation – navigation, for example a HyperLinksButtonGroup

- DialogContent.Main – for the main content

You must put a DialogContentNM control in the DialogContent placeholder that the Dialog control provides.

17.1.8   DialogHeader

A DialogHeader control is a page substructure control. You use it as a substructure for the Dialog page structure controls. It provides these placeholders:

- DialogHeader.Close

- DialogHeader.Navigation

- DialogHeader.Buttons

- DialogHeader.Actions

You must put a DialogHeader control in the DialogHeader placeholder that the page structure control provides.

17.1.9   GlobalFooter

A GlobalFooter control is a page substructure control. You use it as a substructure for any of the Dashboard, List, or Task page structure controls. It provides these placeholders:

- GlobalFooter.FooterInfo

You must put a GlobalFooter control in the GlobalFooter placeholder that the page structure control provides.

17.1.10 GlobalHeader

A GlobalHeader control is a page substructure control. You use it as a substructure for any of the Dashboard, List, or Task page structure controls. It provides these placeholders:

- GlobalHeader.StartButton – you use this for a GlobalLogo control

- GlobalHeader.LoginInfo – you use this for an AccountInformation control

- GlobalHeader.Messaging

- GlobalHeader.Search

You must put a GlobalHeader control in the GlobalHeader placeholder that the page structure control provides.

17.2   Page Structures - Applications

17.2.1   Dashboard

A Dashboard control is a page structure control that you use to create a Dashboard page.

A Dashboard page shows overview information for an entire application or a specific area of an application. Dashboard pages generally do not provide editing capabilities.

The Dashboard control provides these placeholders:

- GlobalHeader
- ApplicationHeader
- ApplicationContent
- GlobalFooter (optional)

17.2.2   List

A List control is a page structure control that you use to create a List page.

A List page generally displays a single list of items. It often has search and filtering capabilities to help users find items. List pages generally do not provide editing capabilities.

It provides these placeholders:

- GlobalHeader
- ApplicationHeader
- ApplicationContent
- GlobalFooter (optional)

17.2.3   Task

A Task control is a page structure control that you use to create a Task page.

A Task page generally shows detailed information about a single item. A task page often provides editing capabilities. Task pages usually do not provide many navigation options, other than the ability to navigate back to the previous page.

It provides these placeholders:

- GlobalHeader
- ApplicationHeader
- ApplicationContent
- GlobalFooter (optional)

17.3 Page Structures - Dialogs

17.3.1 Dialog

A Dialog control is a page structure control that you use to create a single dialog page.

The Dialog control provides these placeholders:

- DialogHeader
- DialogContent

17.4   Page Structures - Framework

17.4.1   SearchResults

A SearchResults control is a page structure control.

It provides these placeholders:

- ApplicationHeader
- ApplicationContentMain

17.4.2   Start

A Start control is a page structure control.

It provides these placeholders:

- ApplicationHeader
- ApplicationMain

Chapter 18

Alphabetical index