Bread Crumb Trail Control 2.0 Requirements Specification

1. Scope

1.1 Overview

Today's websites have complex navigation rules. As a user navigates through a website, it is very easy to become disoriented. A bread crumb trail provides links following the path of the user. This component provides an easily customizable control to provide breadcrumb functionality to a website. The look and feel of the control is set using CSS style sheets.

1.2 Logic Requirements

1.2.1 Control Properties

The control will implement the System.Web.UI.WebControls.WebControl interface with the following required properties:

- PathSeparator – the characters used to separate path nodes
- PathDirection – displays bread crumb from root—to current node or current node to root
- CurrentNodeStyle – the CSS style for the current node
- NodeStyle – the CSS style for all of the standard nodes
- RootNodeStyle – the CSS style for the root node
- PathSeparatorStyle – the CSS style for the path separator text
- MouseOverNodeStyle – the CSS style for the node when a mouse is over it
- DataSource – the data source to bind the control to

1.2.2 Binding Data

The control will allow a user to bind an XML document to dynamically list the trail to display. The design will provide the corresponding XSD for validation.

1.2.3 Pre-defined Sitemap

The control will allow a user to bind an XML data structure containing the entire site map to the control. It will automatically determine what page the user is on and what links to display in the generated trail.

1.2.4 Multiple Parent Nodes

It is possible that multiple pages can point to the same page. The design will provide support for handling multiple parent nodes and automatically detect the correct parent in the generated trail.

1.2.5 Support for Query String Pattern Matching

Target URLs may include query strings with various parameters and values. The design will support query string pattern matching. This is especially relevant to the pre-defined sitemap.

For example, xyz.asp supports two parameters: "id" and "page". So the URL will look like: "http://www.xyz.com/xyz.asp?id=2&page=5". The display name for the page is "XYZ".

In other words, the control will display "XYZ" for all values of "id" and "page".

1.2.6 Dynamic Bread Crumb Trail Generation

Related to multiple parent nodes and query string patterns, the URLs of a web site may be very dynamic. Static XML sitemaps become near impossible to use in some situations. The design will provide a mechanism to dynamically generate the bread crumb trail based on the session state and the pages previously visited by the user.

1.2.7 Support for Visual Studio .NET Designer

The control will support Visual Studio .NET Designer. This will allow the user to drop the control into an ASP.NET page at design time and customize the properties.

1.3 Required Algorithms

The design will provide a reference algorithm for dynamically generating the bread crumb trail.

1.4 Example of the Software Usage

The TopCoder website contains numerous useful pages. However, a user may lose their way while surfing TopCoder.com. In order to orient a user within the website and to provide a way for a user to backtrack a bread crumb trail would be used.

1.5 Future Component Direction

Store the history of URL navigation for various statistical analyses.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

The control will implement the System.Web.UI.WebControls.WebControl interface.

2.1.2 External Interfaces

None.

2.1.3 Environment Requirements

- Development language: C# 1.1
- Compile target: C# 1.1

2.1.4 Package Structure

TopCoder.Web.UI.WebControl.BreadCrumb

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

XML binding needs to be configurable.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

- .NET Framework 1.1

3.2.2 TopCoder Software Component Dependencies:

None.

**Please review the TopCoder Software component catalog for existing components that can be used in the design.

3.2.3 Third Party Component, Library, or Product Dependencies:

None.

3.2.4 QA Environment:

- Windows 2000
- Windows Server 2003

3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

3.4 Required Documentation

3.4.1 Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

3.4.2 Help / User Documentation

- Design documents must clearly define intended component usage in the 'Documentation' tab of Poseidon.