

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits=
    "System.Web.Mvc.ViewPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">
    TreeView
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

    <p>This tree is populated from a hierarchical set of data. However, I used the
        technique of creating a dynamic
        sitemap which is then passed to the Telerik Treeview Extension. The sitemap method
        permits an unknown depth to the tree, whereas
        the hierarchical model requires foreknowledge of the depth. Drag and drop is enabled
        without restrictions - permitting you to
        produce some bizarre hierarchies.
    </p>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="contentLeftHolder" runat="server">
<% Html.RenderPartial("TelerikList"); %>
</asp:Content>

<asp:Content ID="Content4" ContentPlaceHolderID="contentRightHolder" runat="server">
</asp:Content>

<asp:Content ID="Content5" ContentPlaceHolderID="BottomContent" runat="server">
<%Html.Telerik().TreeView().Name("TelEntityTree").HtmlAttributes(new { @class = "teltree"
    })
    .DragAndDrop(true).HighlightPath(true).BindTo("entities", (item, siteMapNode) =>
        {
            item.ImageUrl = siteMapNode.Attributes["imageUrl"].ToString();
            item.HtmlAttributes.Add("value", siteMapNode.Attributes["value"].ToString());
        }
    );
    item.HtmlAttributes.Add("entitytype", siteMapNode.Attributes["entitytype"].
ToString());
    item.HtmlAttributes.Add("title", siteMapNode.Attributes["title"].ToString());
    }

    ).Render(); %>

<div id='roomphoto'>
</div>

</asp:Content>
```