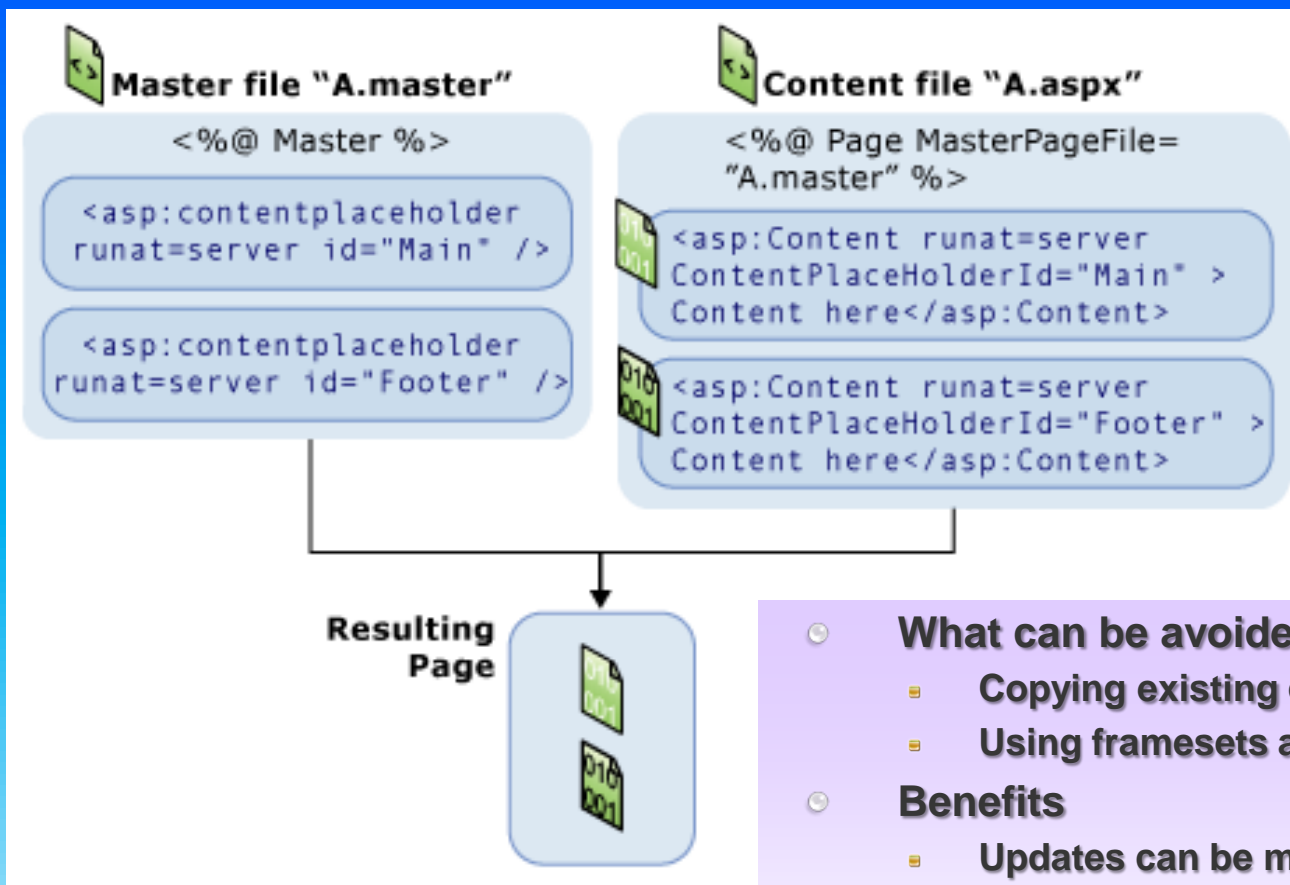


ASP.NET programming

Master page/Content pages
Navigation/SiteMap/Menu control
Passing data in Session variables

Master pages



- **What can be avoided using master pages**
 - Copying existing code
 - Using framesets and include files
- **Benefits**
 - Updates can be made in just one place
 - Easy way of creating one set of controls and code and apply the results to a set of pages
 - Underlying object model allows the master page to be customized from individual content pages

Navigation

- Menu and TreeView controls support the following features
 - In site navigation through integration with the SiteMapDataSource control
 - Programmatic access to the Menu or TreeView object model to dynamically create menus or trees, populate menu items or tree nodes, set properties, and so on
 - Customizable appearance through themes, user-defined images, styles, and user-defined templates
 - Data binding that allows the nodes of the control to be bound to XML, tabular, or relational data
 - TreeView node text that can be displayed as either plain text or hyperlinks
 - Ability to display check box next to each TreeView node

Passing data in Session variables

- Good news
 - Flexible, easy to use
 - Can hold any serializable object
 - Bad news
 - Scalability may be an issue
 - Weak typing, needs casting
 - No Intellisense help in Visual Studio
- } Work-around exists

```
// Page 1
Session["the_variable"] = 25;
// Page 2
int code = (int)Session["the_variable"];
```

How To Use Session Variables in Strongly Typed Manner

- Handler class is created
 - Includes for each session variable strongly typed public static property
 - Session variables are read and written through these properties
- In get accessor
 - For reference types, if session variable is null, either null is returned, or a new object with default properties is initialized and returned, otherwise appropriate type cast session variable is returned
 - For value types default value is returned
- In set accessor
 - Value is saved to appropriate session variable

Strongly Typed Usage of Session Variables

Session handler implementation

```
public class SessionHandler {
    public static String Password {
        get {
            if (HttpContext.Current.Session["Password"] == null)
                return null;
            else
                return (String)HttpContext.Current.Session["Password"];
        }
        set {
            HttpContext.Current.Session["Password"] = value;
        }
    }
    public static int ListIndex {
        get {
            if (HttpContext.Current.Session["ListIndex"] == null)
                return -1;
            else
                return (int)HttpContext.Current.Session["ListIndex"];
        }
        set {
            HttpContext.Current.Session["ListIndex"] = value;
        }
    }
}
```

Session handler
usage

```
if (SessionHandler.Password == null)
    Response.Redirect("Home.aspx");
```

```
SessionHandler.ListIndex =
    ListBox1.SelectedIndex;
```

DEMO

Master page/Content pages
Navigation/SiteMapDataSource
Strongly typed usage of session variables