ViewState

Fritz Onion
DevelopMentor
http://staff.develop.com/onion/

About the speaker

Name	Fritz Onion
Job	Researcher, author, and instructor with DevelopMentor
URL	http://staff.develop.com/onion/
Residence	Wells, ME
Technical Background	C++, COM, ATL, MFC, VB, C#, VB.NET, ASP.NET
Current Activity	Completing 'Essential ASP.NET' to be published by Addison Wesley
Teaching	Essential ASP.NET, Guerrilla .NET, Essential .NET, C#
Speaking	Here, C# Summer Camp @ MS (August), WinDev '02 @ Boston (October)
Favorite color	blue

ViewState is...

- **Used to retain control state between post-backs**
  - How much it is used (and if at all) depends on the control
- **A generic client-specific state repository**
  - Accessible through the Page.ViewState property
- **A collection of state stored in a hidden field**
  - The field is always

```
<input type=hidden name='__VIEWSTATE' value='...'/>
```

- **Persisted as a base64-encoded string**


```
dDwyMTE1NjE3OTg4O3Q8cDxsPENhcnQ7PjtsPGw8Pjs+Pjs7Pjs+7Yi0dU0f/WiU7Yoble4ERMYzcnk=
```


- **The solution to bringing the control-based programming model to WebForms**
  - Enables state retention for controls

ViewState is not...

- **Necessary if you are not making use of state retention**
  - If you populate a control's state on each page access
- **Cheap**
  - Base64 string encoded state using tuple-data structure
- **Necessary if you are not posting back to the same page**
  - Only works if the form post action is the same page
- **Well understood :)**

Disabling ViewState

- **You can disable ViewState at the page level if the page doesn't post back to itself**

```
<%@Page EnableViewState='false' %>
```

- **You can disable ViewState at the application level if none of your pages post back to themselves**

```
<!-- inside web.config for your app -->
<Pages EnableViewState='false'/>
```

- **You can disable ViewState at the machine level if you want to break 90% of the ASP.NET applications on that machine**

```
<!-- inside machine.config -->
<Pages EnableViewState='false'/>
```

- **You can disable ViewState at the control level if you are not handling that control's events and the control's state is reset on each request**

```
<asp:DataGrid EnableViewState='false' .../>
```

ViewState usage by controls

Control Class	Uses ViewState?
HtmlImage	No
HtmlInputFile	No
HtmlInputHidden	No
HtmlInputImage	No
HtmlInputRadioButton	No
HtmlInputText	No
HtmlInputButton	No
HtmlInputCheckBox	No
Xml	No
Repeater	Yes
HtmlForm	Yes
HtmlGenericControl	Yes
HtmlSelect	Yes
HtmlTable	Yes
HtmlTableCell	Yes
HtmlTableRow	Yes
HtmlTextArea	Yes
HtmlAnchor	Yes
HtmlButton	Yes
All WebControl-derived controls	Yes

ViewState as a State Repository

- **Much like Session State or Cookies, ViewState can act as a client-specific state repository**
  - Accessible through the Page.ViewState property
  - Can store any data, as long as it is serializable
  - State is only retained when posting back to the same page
  - State is round-tripped (like cookies)
  - State is not protected (by default)

Client-specific state comparison

Type of state	Scope of state	Advantages	Disadvantages
Session	Per client	? Can configure to be shared across machines in a web farm	? Requires cookies or URL mangling to manage client association ? Off-host storage can be inefficient
Cookie	Per client	? Works regardless of server configuration ? State stored on client ? State can live beyond current session	? Limited memory (~4kb) ? Clients may not support cookies or may explicitly disable them ? State is round-tripped with each request
ViewState	Across POST requests to the same page	? Works regardless of server configuration	? State is only retained with POST request made to the same page ? State is round-tripped with each request

Decoding ViewState

- **ViewState is persisted as a tuple-based data structure and encoded as a base64 string**

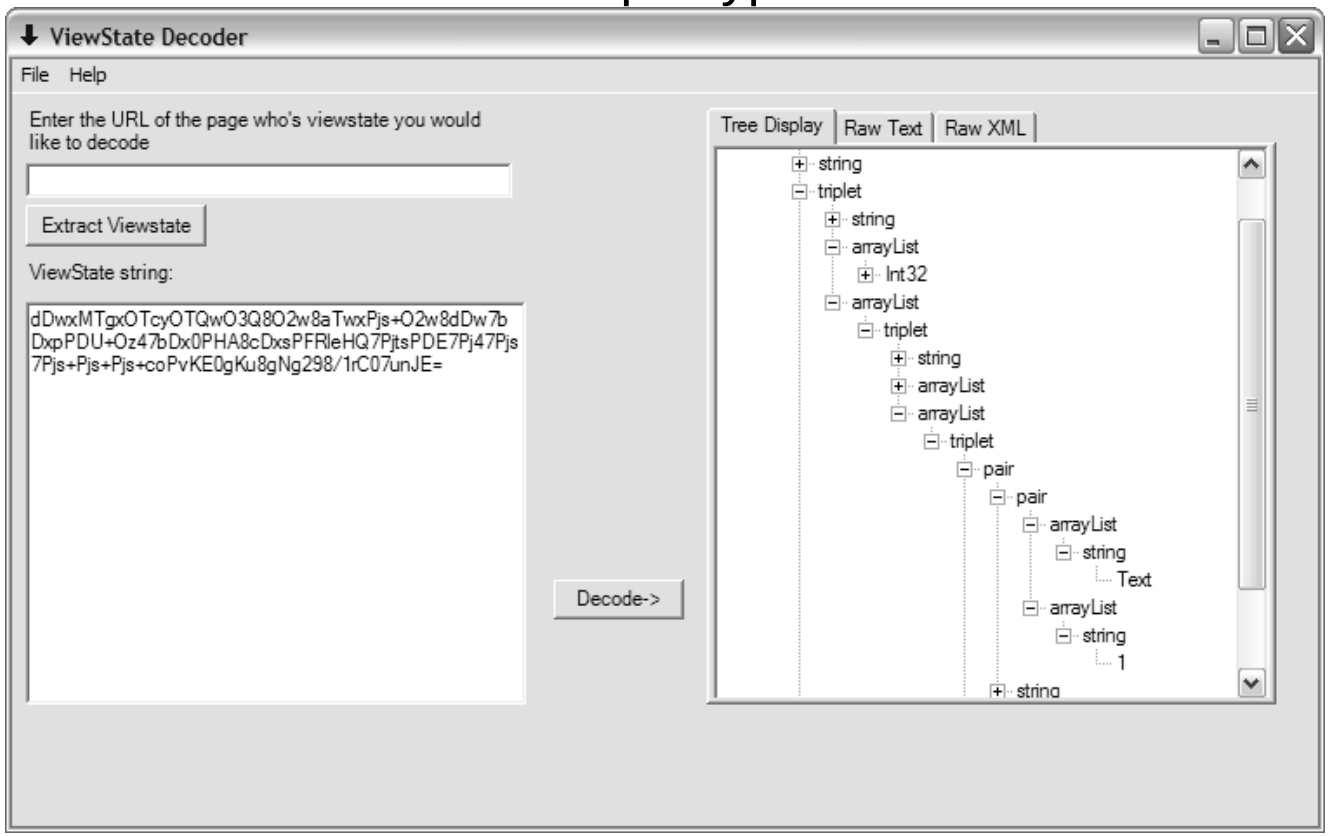
dDwxMTgxOTcyOTQwO3Q8O2w8aTwxPjs+O2w8dDw7bDxpPDU+Oz47bDx0PHA8cDxsPFR leHQ7PjtsPDE7Pj47Pjs7Pjs+Pjs+Pjs+coPvKE0gKu8gNg298/1rC07unJE=

↓

t<1181972940;t<;l<i<1>;>;l<t<;l<i<5>;>;l<t<p<p<l<Text;>;l<1;>>;>;;>;>>;>>;>r(M * 6kN?

ViewStateDecoder

- **Available at**
  - http://www.develop.com/devresources/resourcedetail.aspx?type=t&id=827



Protecting ViewState

- **ASP.NET provides two mechanisms for protecting ViewState**
  - Tamper-proofing using a hashcode (SHA1 or MD5)
  - Encryption using Triple DES symmetric algorithm (3DES)
- **Tamper-proofing can be enabled at the Page level**

```
<%@ Page enableViewStateMac='true' %>
```

- **Or at the application or machine level**

```
<!-- in web.config or machine.config -->
<pages enableViewStateMac='true' />
```

ViewState Encryption

- **Must be enabled at the machine level:**

```
<machineKey validation='3DES' />
```

- **For deployment on a WebFarm, you must specify the same validation key on all machines:**

```
<machineKey validation='3DES'
   validationKey='7DFB068D0FC3D9E2E3C2BCA32DF9509DEFF71B8F44DF95578CC
   FCD2B6EA50475BEE615AF0E64072EC07A1D4720F11A5125947C592512BA7229081
   2F911963CC8' />
```

Generating a Validation Key

```
using System;
using System.Text;
using System.Security;
using System.Security.Cryptography;

class App {
  static void Main(string[] argv)  {
        int len = 48;
        if (argv.Length > 0)
          len = int.Parse(argv[0]);

        byte[] buff = new byte[len/2];
        RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider();
        rng.GetBytes(buff);
        StringBuilder sb = new StringBuilder(len);
        for (int i=0; i<buff.Length; i++)
          sb.Append(string.Format("{0:X2}", buff[i]));

        Console.WriteLine(sb);
  }
}
```

Summary

- ViewState is used by controls to retain state across post-backs
- ViewState is not always necessary, depending on control usage
- ViewState can be disabled
- ViewState is stored as a base64-encoded string
- ViewState can be protected against tampering
- ViewState can be encrypted