



ViewState

The good, the bad and how to fix it.

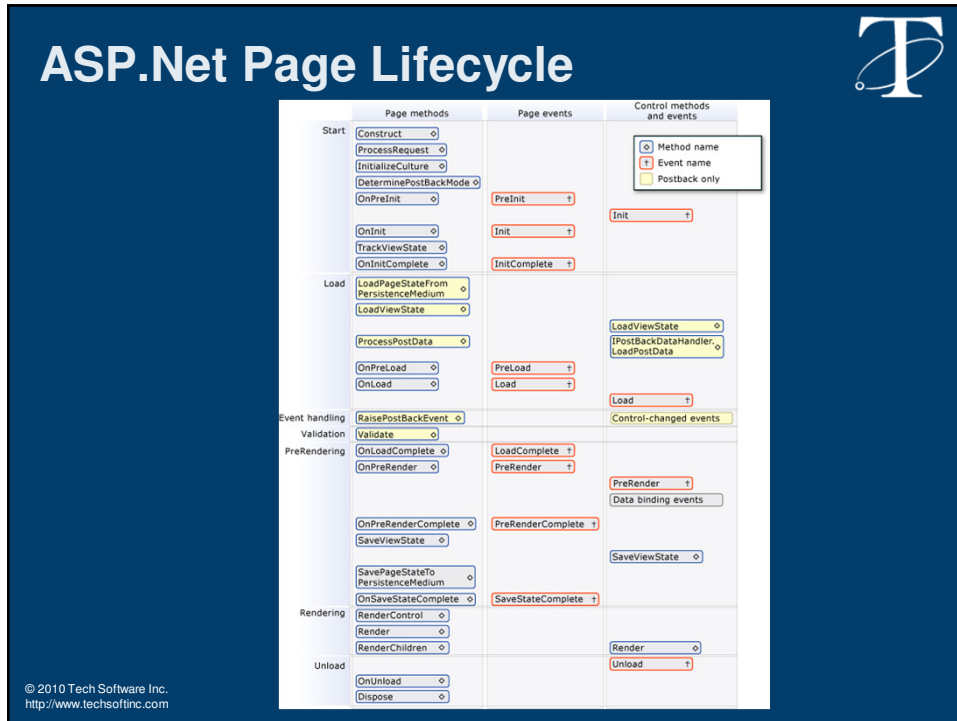
© 2010 Tech Software Inc.
<http://www.techsoftinc.com>



What's the problem?

- The web has no state
 - Server sees each request as a totally new conversation
 - Session State attempts to solve some of this for a “session”
 - But what about a page request?
 - Form Fields part of the solution
 - Font? Literals? Other non-form-field based values?
- Arguably a “leaky abstraction” of WebForms
 - OK, maybe true
 - But it gets the job done!

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>



All Controls Have ViewState

- ... and Page is a control!
- Any object can be added to ViewState
- The ViewState of a control is the view state of that control AND the collection of view states of all its child controls

© 2010 Tech Software Inc.
http://www.techsoftinc.com

Where do I see ViewState?



- Simple! The `__VIEWSTATE` form field
- Encoded as a string and “stored” as hidden field
 - Base64 representation of two state bags
 - Control State and true View State
 - Can be quite large

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

How do things get in ViewState



- For a control, the control developer puts it there.
- Typical MS (and other) control uses ViewState as the backing-store for property values
- Typical “class” Property

```
string _text;

public string Text
{
    get { return _text; }
    set { _text = value; }
}
```

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

How do things get in ViewState



- Typical “Control” Property

```
//Note No Backing Field
public string Text
{
    get {
        return ViewState["Text"] as string
            ?? "Default Text"; }
    set { ViewState["Text"] = value; }
}
```

- Note “correct” handling of default value
 - Important for dealing with declarative (ASPX) values

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

What’s typically in it?



- Well, that’s the problem. All sorts of stuff
- TextBox values IF you’re change tracking
 - An OnTextChanged handler
- (Almost) any value you set in code that’s not represented in a form field
 - Create the control, set the properties, THEN add to the control collection and you’re ok
- DataSets!
 - Grids are famous for this, storing the data used to load the grid to be “helpful”

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

But you can add to it too



- Page developer can add values to ViewState simply enough:

```
ViewState["SortDirection"] = "ASC";
```

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Where's it come from?



- In the SaveViewState() method (after the PreRenderComplete event)
- ASP.Net gathers the view state for the page.
- The page's view state made up of its own view state plus its controls' view state
 - Each control's view state made up of its own view state plus its controls' view state
 - Each control's view state made up of its own view state plus its controls' view state
 - Each control's view state made up of its own view state plus its controls' view state
 - › Each control's view state made up of its own view state plus its controls' view state

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

And what happens with it?



- On a postback, after InitComplete ASP.Net deserializes ViewState and uses it to set control property values back to their saved values
- The mapping is done based on their position in the control tree – IMPORTANT!
 - If controls are out of position... boom!
- Option to enable mapping by control id, (discovered in reflector) but this is a control-developer option, not a runtime option (disfortunate!)

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Controls out of position?



- Sure, two ways.
 - Change the page (new version) after you send the client the page but before he sends it back
 - Move (add) the control in code the wrong way or at the wrong time.
 - Add controls in page init
 - Remember, no view state tracking until after init
 - Add controls at the END of the controls collection
 - An extra node at the end of a branch is never seen
 - If I need VS for control 1, 6, and 22 and I have it for 1, 6, 22 and 37 that's ok.
 - If I have it for 1, 4, 6, and 22... boom!

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

What if I don't want ViewState



- Pre .NET 4.0
 - EnableViewState="false"
 - Great solution, stops the view state from being serialized
 - Problem is it's per control
 - Lots of controls on a page to set
 - If a control doesn't save view state it doesn't save its children's view state
 - Set to false on Page → NO View state at all on that page

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

.NET 4 improvements



- Addition of ViewStateMode Property on a control
 - Inherit - Inherit the value of ViewStateMode from the parent Control.
 - Enabled - Enable view state for this control even if the parent control has view state *disabled*.
 - Disabled - Disable view state for this control even if the parent control has view state *enabled*.
- Allows you to Disable ViewState at a global level and Enable it on only the one or two controls that need it

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

.NET 4 and EnableViewState



- EnableViewState still exists
 - And “false” TRUMPS ViewStateMode
 - If you disable view state (EnableViewState=false) there is NO view state, period, full stop, end of story, don't care what ViewStateMode says
- If EnableViewState = True _then_ ViewStateMode counts.
- Greenfield development best practice is ViewStateMode=Disabled at top level and enabled on controls as needed.

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

What is ViewState (technically)



- The ViewState property on a control is a StateBag from System.Web.UI
- Similar to a HashTable
 - But with change tracking
- ViewState Values are stored in this StateBag
 - Values are stored within a StateItem which is a pair of Bool and Object, (IsDirty and Value), one item per key

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

How is ViewState Save to Page?



- As a Base64 String
- ObjectStateFormatter class “serializes” the ViewState
- Optionally (on by default) adds a MAC (Machine Authentication Check) to prevent tampering
 - Be aware of this for web farms
 - MAC Key must be the same on all servers in farm
 - Change in machine.config
- Optionally (off by default) encrypts
 - Again, in web farm be aware of common key

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

ObjectStateFormatter



- Responsible for turning ViewState and ControlState object graphs into a string
- OSF designed to do one thing very well
 - Serialize ViewState TO A STRING
 - (Luckily) Does this as a two step process
 - Designed to minimize length of serialized representation for common types used in ASP.Net Controls
 - IndexedString serializes 255 “most common” strings in just 2 bytes

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

LosFormatter



- Limited Object Serialization Formatter
 - Original ViewState formatter
 - Many web references still refer to it as the formatter used to serialize ViewState
- Rendered obsolete with introduction of ObjectStateFormatter in .NET 2.0
- Methods are simply wrappers around calls to replacement methods in ObjectStateFormatter
- PageStatePersister no longer uses LosFormatter at all

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

ControlState



- Added in ASP.Net 2.0 when MS recognized that sometime you *need* ViewState info
- You can turn off ViewState, you can't turn off ControlState
- Controls store only bare minimum information in ControlState
 - If you won't work AT ALL without it then it's control state
 - If you won't work well, but will work, then it's view state

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

TechVSE (ViewStateEliminator)



- Set out to solve two of the problems with ViewState
 - Round-trip payload size
 - Performance improvement, faster to move w/out VS
 - Bandwidth utilization improvement, fewer bytes paid for
 - Improve Security
 - If it's not there, they can't hack it!
 - You shouldn't store secrets in VS, but...
 - Can be partially addressed w/encryption, but at a performance expense

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

General Requirements



- Offload ViewState storage from hidden field to SQL Database
- Save ViewState as blob in DB and use a Guid as Key
 - Then just Guid is sent back and forth to client
- Don't require any code changes to application
 - One common approach is to derive your own Page class and then have all your pages derive from that.
 - Not a great plug-and-play solution!
- Report on ViewState sizes and Offenders!

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Staged implementation



- Round 1 – Replace ViewState persistence with our own
- Round 2 (because I'm anal) – Use the __VIEWSTATE field in the ASP.Net form, don't add another one.
- Round 3 – Good reporting

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

App_Browsers



- Common directory ASP.Net understands that contains files about different browser types.
- One of the things you can do is to override default implementations of functionality
 - This is how many controls handle native and “mobile” representations of their markup
- Override is handled through a ControlAdapter
 - Allows you to “adapt” the processing of a control for a specific browser
 - PageAdapter exists as a specialized ControlAdapter
- Default.Browser applies to ALL browsers

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

PageAdapter Lets you...



```

DeterminePostBackMode()
GetPostBackFormReference(string)
GetRadioButtonsByGroup(string)
GetStatePersister()
PageAdapter()
RegisterRadioButton(System.Web.UI.WebControls.RadioButton)
RenderBeginHyperlink(System.Web.UI.HtmlTextWriter, string, bool, string)
RenderBeginHyperlink(System.Web.UI.HtmlTextWriter, string, bool, string, string)
RenderEndHyperlink(System.Web.UI.HtmlTextWriter)
RenderPostBackEvent(System.Web.UI.HtmlTextWriter, string, string, string, string)
RenderPostBackEvent(System.Web.UI.HtmlTextWriter, string, string, string, string, string, string)
RenderPostBackEvent(System.Web.UI.HtmlTextWriter, string, string, string, string, string, string, bool)
TransformText(string)
CacheVaryByHeaders
CacheVaryByParams
ClientState
  
```

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Did you notice?



- GetStatePersister?
 - How nice of MS to provide an exit for us!
- Allows us to “override” how the [View]State is persisted.

```

public class PageAdapter :
System.Web.UI.Adapters.PageAdapter
{
public override PageStatePersister
GetStatePersister()
{
return new SQLPageStatePersister(Page);
}
}
  
```

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

SQLPageStatePersister



- Derives from PageStatePersister
- Overrides Save() and Load()
- Default (MS) implementation of Save()

```
public override void Save()
{
    if (base.ViewState != null ||
        base.ControlState != null)
    {
        this.Page.ClientState =
            Util.SerializeWithAssert(this.StateFormatter, new
                Pair(this.ViewState, this.ControlState));
    }
}
```

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Saving ViewState to SQL



- new Pair(this.ViewState, this.ControlState)
- The ObjectStateFormatter rolls this object graph up into a binary representation and `_then_` serializes it to a string, base64.
 - Luckily those are different methods
 - We simply call the binary serialization and then store the binary blob in sql
- Not only are we not sending it down the pipe, but we're not wasting time/size on base64!
- Base64 will be 1/3 again larger

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

This seems simple enough...



- Just replace their Save() with our Save() to SQL and put the Guid key into this.Page.ClientState
- One small problem...

```

internal string clientState
{
    get
    {
        return this._clientState;
    }
    set
    {
        this._clientState = value;
    }
}

```

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Thus two stages...



- Stage one, we added our own form field to contain the Guid (ViewStateGuid)
- It works!
- But now we have an additional field in the collection, and anyone doing “special” processing with __VIEWSTATE won’t work
 - Like Sending it back on an Ajax Request

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Step 2, can we set an internal?



- C# “rules” say no, it’s not public
- But reflection says otherwise!
- Reflection allows you to spelunk the entire class including internal AND private!
- But reflection is S L O W (relative term)

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Reflection and IL Generation



- Use reflection to get references to the Setter of ClientInfo

```
Type pageType = typeof(Page);  
  
MethodInfo set =  
pageType.GetProperty("ClientState",  
BindingFlags.Instance |  
BindingFlags.Public |  
BindingFlags.NonPublic).GetSetMethod(  
true); // True = non-public
```

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Now we can set that internal!



- Write our own “override” of that setter
 - A new method on the Page class that calls the internal setter
- Do the same thing to load back the ViewState
 - “override” RequestViewStateString property getter

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

“Override” Code



```
private delegate void ClientStateSetter(Page This, string
Value);

//Name, ReturnType, Params, Owing Type
DynamicMethod dm = new
DynamicMethod("_DynamicSet_ClientState", null, new Type[] {
pageType, typeof(string) }, pageType);

ILGenerator ilg = dm.GetILGenerator();
ilg.Emit(OpCodes.Ldarg_0);
ilg.Emit(OpCodes.Ldarg_1);
ilg.Emit(OpCodes.Callvirt, set);
ilg.Emit(OpCodes.Ret);

ClientStateSet =
(ClientStateSetter)dm.CreateDelegate(typeof(ClientStateSett
er));
```

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

And as a result



- ```
<input type="hidden"
name="__VIEWSTATE" id="__VIEWSTATE"
value="{afbc71d9-9d0d-4c3d-9e99-
d54be7575f2e}" />
```
- <http://www.dayget.com.au/>

© 2010 Tech Software Inc.  
<http://www.techsoftinc.com>

## Reporting on ViewState



- Now it's in a database we have reporting options!
- Still a work in progress to provide "nice" reporting options (web pages, charts, graphs, etc.)
- ```
select avg(dataLength(viewState))
from ViewStateStore
```

 - Our avg viewstate (binary) is 10K

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

More Reporting



- `select case when charindex('?', url) > 0
THEN substring(url, 0, charindex('?', url))
ELSE url END as URL, datalength(ViewState)
as size from ViewStateStore order by size
desc`

~/Admin/ViewNotificationLog.aspx	1727642
~/Admin/ViewNotificationLog.aspx	1727642
~/Admin/ViewNotificationLog.aspx	1611346
~/Admin/Study/StudySiteStatusAsOf.aspx	1104097
~/SearchResults.aspx	1024244
~/SearchResults.aspx	1023930
~/Reports/Events.aspx	764246
~/SearchResults.aspx	734732

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Licensing TechVSE



- Want to license this solution for sale
- Need to support time and URL licensing
- Don't want to build custom assemblies each time
- XML to the rescue, but want to have it signed

© 2010 Tech Software Inc.
<http://www.techsoftinc.com>

Create “License” object



- Simple C# class

```
[Serializable]
public class VSISLicense
{
    public string Licensee;
    public DateTime Expires;
    public List<string> AuthorizedHosts;
}
```

- Can be simply serialized as XML

© 2010 Tech Software Inc.
http://www.techsoftinc.com

But how do we secure it?



- .NET BCL to the rescue – The SignedXML class

```
RSACryptoServiceProvider csp = new
RSACryptoServiceProvider();
csp.FromXmlString("<RSAKeyValue><Modulus>MyPUBLICKey</Modu
us><Exponent>AQAB</Exponent></RSAKeyValue>");
XmlDocument xmlDoc = new XmlDocument();
xmlDoc.Load("PathToLicenseFile");
XmlNode dsig = xmlDoc.GetElementsByTagName("Signature",
SignedXml.XmlDsigNamespaceUrl)[0];
SignedXml sxml = new SignedXml(xmlDoc);
sxml.LoadXml((XmlElement)dsig);
//If not valid return w/out setting LicenseLoaded to true
if (!sxml.CheckSignature(csp))
{
    log.Error("License signature invalid.");
    return;
}
// Now simply deserialize the XML
```

© 2010 Tech Software Inc.
http://www.techsoftinc.com