



Going Outside the Lines: Building Web Services Using System.Xml

Tim Ewald

tewald@microsoft.com

Motivation

- Web services are really about XML
- Plumbing tends to convert XML to objects
 - Appeals to many who remain angle-bracket impaired
 - Power of XML is lost
- How can you build Web services that use XML *and* expose it to the developer?

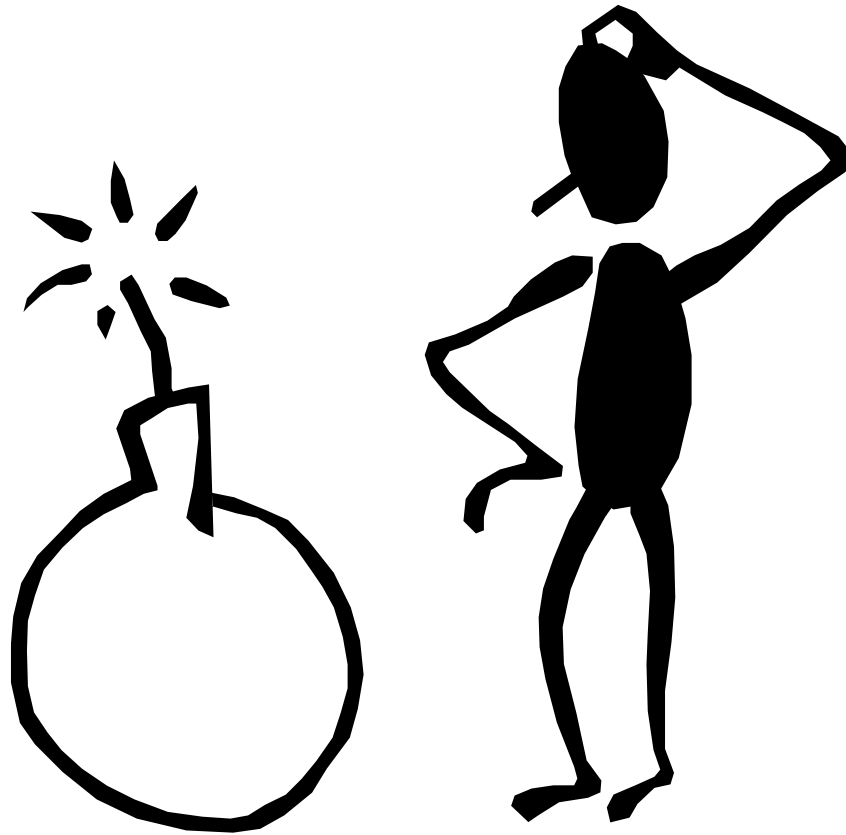
Alternatives

- Low-level .NET Framework APIs
- Low-level WSDK APIs
- Bending WebMethods to your will

Low-level .NET APIs

- Can implement Web services using low-level .NET APIs directly
- System.Xml for SOAP message work
 - Combination of streaming/DOM works best
- System.Net for client-side Web access
- System.Web for server-side HTTP support
 - IHttpHandler is your friend!

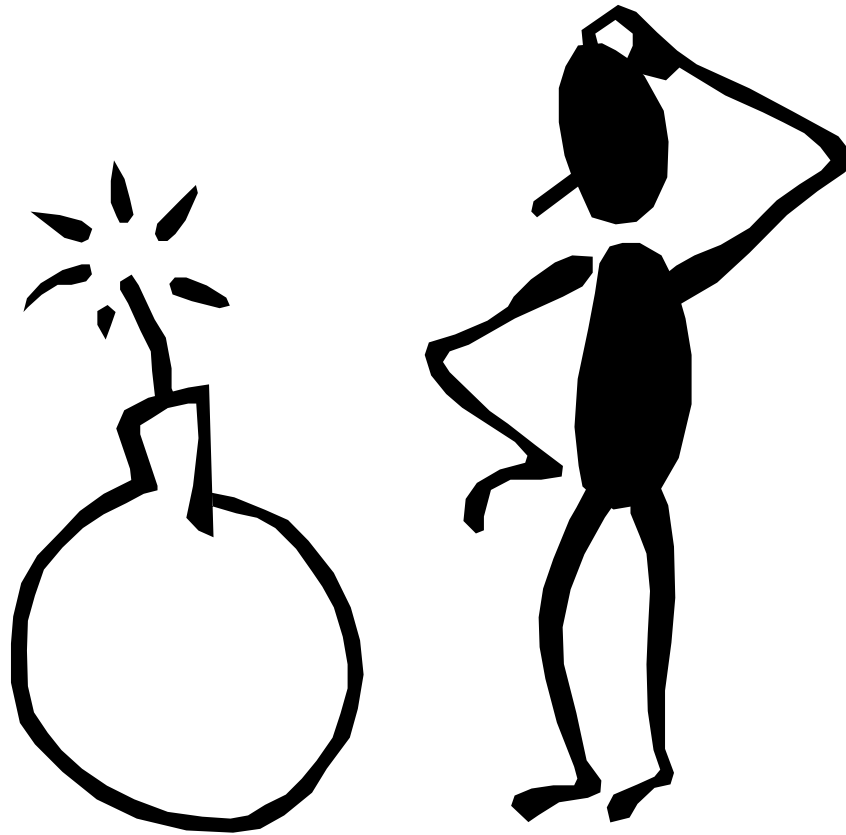
Example



WSDK in the Raw

- The Web Service Development Kit Tech Preview offers simple APIs to help
- SoapEnvelope class ensures messages have correct format
- SoapWebRequest/Response

Example



Extreme WebMethods

- Scratch built solutions are fun
- WebMethod integration makes sense
 - Allows choice of programming models
 - Allows use of others extensions
 - Better tool support
- But WebMethods need a few “adjustments”

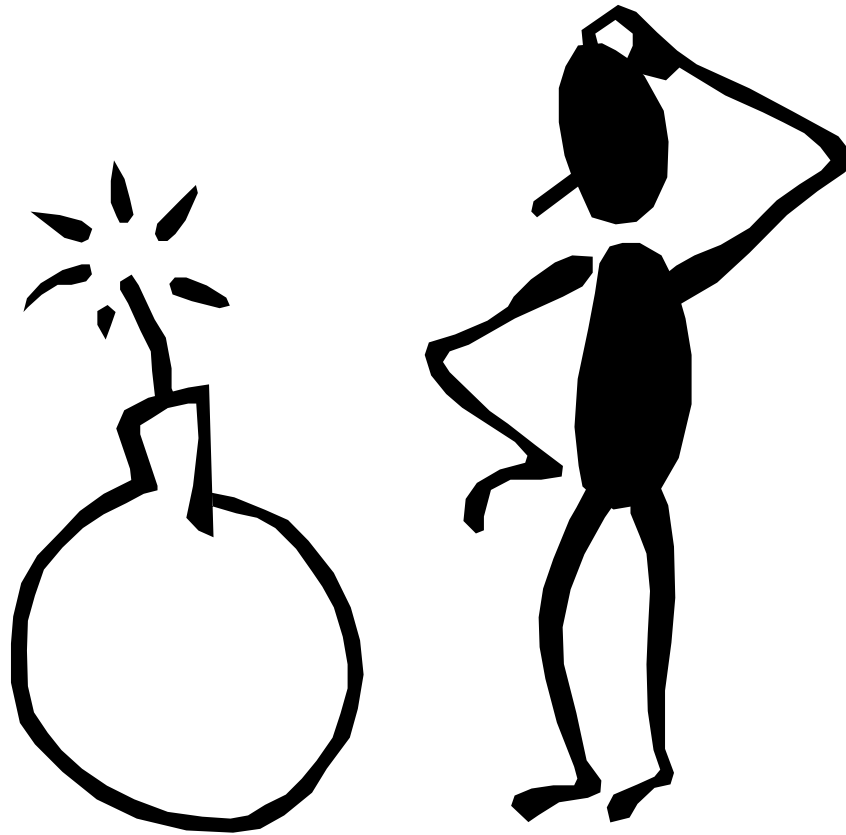
3 Pieces of WebMethods

- WebMethods plumbing really does three things...
 - Message dispatching
 - WSDL generation
 - Parameter marshaling

WSDL generation

- WSDL generation not mandatory
 - Provided for convenience
 - Not used for message dispatch
 - Can be disabled via config file
- Can write more sophisticated WSDL
 - Offers more flexible solutions for versioning, etc.

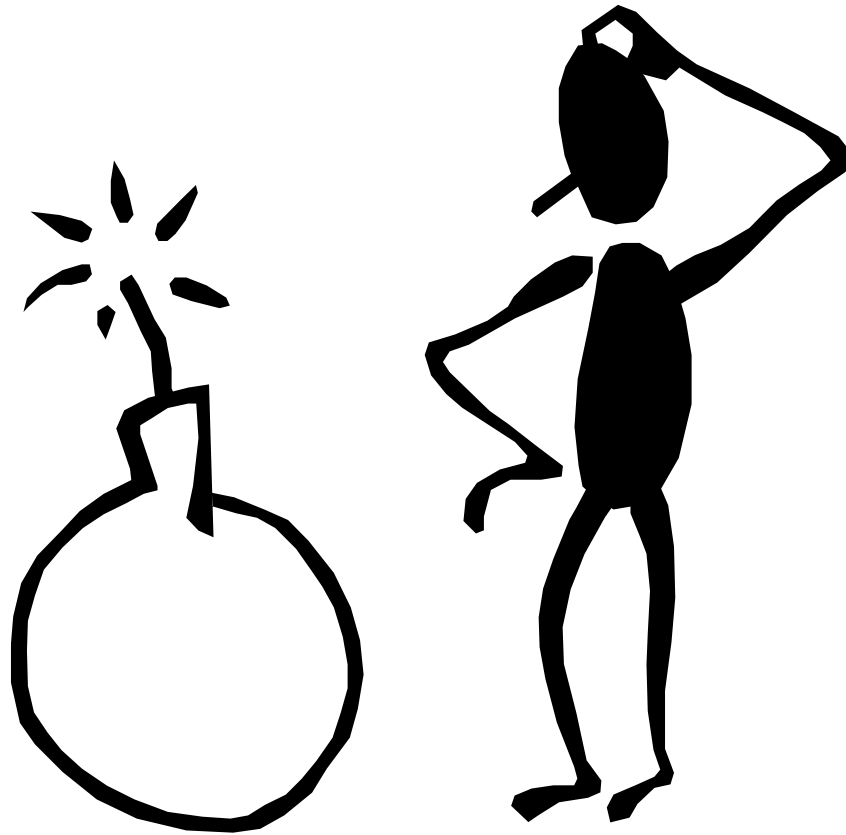
Example



Parameter Marshaling

- Parameters marshaled to CLR types by default
- Can access parameters as XmlElements as desired
 - May need odd XML to achieve desired result
- Can access entire message as XmlElement
 - With some restrictions

Example



Parameter Marshaling

- Can expose SOAP messages as streams if desired
- Requires use of SOAP extension that plays games with message buffers
 - No more expensive than any other extension
- Minor complexity intrinsic in handling faults

Example



Summary

- ASP.NET provides plain vanilla support for building Web services using objects
- Can build your own XML-centric services using low-level .NET and WSDK APIs
- Can mix low-level XML APIs with WebMethods for pleasant blend of convenience and power

References

- MSDN XML Web Services Developer Center
 - <http://msdn.microsoft.com/webservices>
- Presentation and source code available at my web site
 - <http://www.gotdotnet.com/team/tewald>