

Defining Development Standards for Reusable User Interface Components

Max Mulawa, Rich Picking and Vic Grout

Centre for Applied Internet Research (CAIR), University of Wales, NEWI, Wrexham, UK
max.mulawa@gmail.com {r.picking|v.grout}@newi.ac.uk

Abstract

This paper describes a standard for development of ASP.NET user-interface components. This standard aims to assess the quality of implemented components. The standard covers many issues connected with web component development such as accessibility, cross-browser support, rich design-time support, API usability, globalization, localization, deployment. The paper provides an analysis of standard structure and ideas behind the standard.

Keywords

.NET Components, Web standards, ASP.NET, Web Custom controls

1. Introduction and Background

Fast development of the software industry causes a need for the creation of standards, which would increase the quality of application and components. Standards in web development are of increasing importance as programmers strive to make their components work across all browsers and accessible by the broadest set of customers (Clary, 2003). This Standard is approaching one the areas of web development and is aiming to formalize the structure of ASP.NET UI components.

Currently, there are no standards defined for the reusable user interface (UI) web components development process. Much of the information connected with this subject has been published either as technical recommendations or reviews of the web technologies used to develop UI components.

1.1 Why standards are necessary?

Programming standards are important to programmers and consumers for a number of reasons. Such a standard:

- Reduces time of learning component interface and functionality.
- Improves quality of produced components.
- Promotes some proven design principles.
- Improves team collaboration during development process.

A comprehensive standard is essential for successful product delivery. The standard helps in enforcing best practices and avoiding common pitfalls and makes knowledge dissemination across the team easier (Doomen, 2003).

2. Structure and Scope of the Standard

The ASP.NET UI Component Standard specifies rules for development of web UI reusable components and tries to encourage component developers to follow best practices. The standard evaluates results of the development but also need to be considered during that process.

The purpose of the standard is to assure a high quality for the reusable web UI components. The standard consists of rules, which are divided into categories and subcategories. Each rule has its own scale and weight so that not every rule is of the same importance. The standard allows different levels of compatibility and evaluation of the rules affects the whole process of development.

Rules for this standard specify:

- Web standard compliance of the component and component support for standard aware development.
- Constraints on API of the component.
- Documentation structure.
- Integration of the component with IDE.
- Structurised support of CSS, JavaScript by the component.
- Globalization and localization support.
- Structurised deployment process.

This ASP.NET UI Component development does not specify:

- Methodology for development process of the component.
- Coding standards for .NET languages like VB.NET or C#

In a similar way to making the standard usable, this document requires standards in order to ensure clarity. Certain conventions are used throughout this document to add emphasis and improve readability. Below are some of the common conventions used throughout this document.

2.1 Standard Structure

The whole standard is divided into section and subsections. Each section/subsection starts with an overview of the rules and recommendations presented in the section. The overview is followed by tabular list of rules in this section. This tabular representation could be used as a checklist by a Standard user.

The checklist table has the following form:

The columns are:

Column Name	Description
Rule	Number and text of the rule or recommendation.
Validation	Indicates method of verification of the compliance.

2.1 Rule Structure

Each rule in the Standard has following structure:

Item	Description
Rule/Rec.	Contains content for the rule/recommendation.
Category	The category of the rule.
Code	Unique code assigned to the rule
Relative importance	The rating allows the Standard user to quickly ascertain which rule/recommendation has the greatest impact on the success of a component. Relative importance rating will be based on defined scale (very important (3 stars), important (2 stars), less important (1 star)). To determine the 'Relative Importance' of each rule the feedback from the ASP.NET community and experts is needed.
Strength of evidence	<p>Determines supporting evidence and experts opinions on that rule. Strength of evidence rating is based on defined scale (Strong Research Support (3 stars), Weak Research Support (2 stars), None (1 star)).</p> <p>Strong Research Support</p> <ul style="list-style-type: none"> • Cumulative and compelling, supporting research-based evidence • No known conflicting research-based findings • Expert opinion agrees with the research <p>Weak Research Support</p> <ul style="list-style-type: none"> • Limited research-based evidence • Conflicting research-based findings may exist and/or • There is mixed agreement of expert opinions <p>None</p> <ul style="list-style-type: none"> • No research-based evidence • Limited or conflicting expert opinion
Preconditions	Conditions that need to be met before applying the rule.
Rationale	Explains the thoughts and purpose behind a rule.
Tips	Precedes text used to illustrate a rule or recommendation. It also provides advice or guidance to implementers or programmers.
Exceptions	Explains the situation when following a rule would be inappropriate.
Validation	Describes the method of verification of compliance.
Platform	Specifies the technology platform that a rule can be applied. Possible values: ASP.NET, ASP.NET 1.*, ASP.NET 2.0.
Related Rules	Rules that are related to the current rule.
Sources	Publications that are source of the rule, or that provide supporting evidence for the rule.

The source of *Strength of evidence* and *Relative importance* ratings is Balley (2003).

A rule should be broken only for compelling reasons where no reasonable alternative can be found. The author of the violating code shall consult with at least one knowledgeable colleague and a senior designer to review said necessity. A comment in the component documentation explaining the reason for the violation is mandatory.

Recommendations are not a comprehensive academic theory that has strong predictive value; rather they should be prescriptive, in the sense that they prescribe practice with useful sets of DOs and DON'Ts. Recommendations should be presented with justifications and examples.

2.2 Standard Sections

Many of the rules and recommendations were taken from:

- Web Content Accessibility Guidelines 2.0
- IBM Accessibility Checklist
- Section 508 document
- W3C XHTML 1.1 Specification
- MSDN Design Guidelines for .NET Class Library Developers.
- IEEE standard for user documentation
- Technical publication on ASP.NET technology (Esposito,2006; Kothrari, 2003; Muhamad 2003; Cameron 2003; Abrams 2005)

The Standard is divided into the following major sections:

a) **Web standard compliance (Accessibility, Component markup)**

This section contains the rules for ASP.NET UI components to support accessibility features. "Accessible" means usable to a wide range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning difficulties, cognitive limitations, limited movement, speech difficulties, and others (Caldwell, 2005).

Following these rules and recommendations will also make the markup produced by the component more accessible to the vast majority of web users, including older users. It will also enable users to access component markup using many different devices - including a wide variety of assistive technologies. It is important that Web Custom Controls emits HTML that is conformant with XHTML standards as discussed by Walther (2005).

The benefits of Component markup section compliance are listed below:

- Cleaner markup produced by component and better user agent interoperability.
- Valid component markup reduces errors in composition and rendering phases.
- Pages that will use components will be more accessible and usable.
- Enables easier transition to XML-centric world.
- Using valid component markup helps to make pages conform more readily to accessibility standards.

Valid component markup is much easier to read programmatically for situations in which markup is processed by a computer instead of being read by users, and the document can be manipulated using transformations.

b) API Design, CLS Compliance, User documentation

Developer productivity can be seriously hampered by inconsistent design. These non-conforming components will function, but not to their full potential as suggested by Abrams (2005). Reasons why Component API (Application Programming interface) Usability is so important are listed below:

- Increases learnability of your component interface
- Reduces errors that developers can make
- Helps to make better design decisions for component developer, which causes maintenance costs reduction.
- Increased developer productivity
- Decreases training and support costs
- Increased customer satisfaction

Targeting the CLS is an excellent way to ensure cross-language interoperation. ASP.NET web custom control designers can use the CLS to guarantee that their APIs are callable from a wide range of programming languages; therefore, component is likely to have a wider customer base than a non-CLS-compliant version would. CLS rules apply only to those parts of a type that are exposed outside the defining assembly. CLS rules do not apply to internal implementation within an assembly. A type is CLS compliant if all its publicly accessible parts (those classes, interfaces, methods, fields, properties, and events that are available to code executing in another assembly). See Microsoft (2003) for more details.

Anyone who uses components needs accurate information about the correct way to use it. If the information is supplied in a convenient form and is easy to find and understand, the users can quickly become proficient at using the product. Well-designed documentation not only assists the user and helps to reduce the cost of training and support, but also enhances the reputation of the product, its producer and its suppliers.

c) Design-time support

The .NET Framework was written with design-time support in mind. Design-time support it's the ability to connect several components together without writing code. The .NET Framework was written to support very rich design-time capabilities while allowing this design-time specific code to reside in a separate assembly so it does not contribute to the size of the runtime library (Duthie, 2004).

By adding properties and events, you can extend the value of controls and make them easier for developers to work with in an integrated development environment (IDE). Opening opportunities for developers to work with visual design tools promotes control reuse and ultimately enhances productivity.

d) Cross-Browser support

Browser Support does not mean that everybody gets the same experience. Expecting two users using different browser software to have an identical experience fails to embrace or acknowledge the heterogeneous essence of the Web. In fact, requiring the same experience for

all users creates a barrier to participation. Availability and accessibility of content should be our key priority.

An appropriate support strategy allows every user to consume as much visual and interactive richness as their environment can support. This approach commonly referred to as progressive enhancement, builds a rich experience on top of an accessible core, without compromising that core as suggested by Koechley, 2006.

e) ASP.NET infrastructure

ASP.NET UI Component (web custom control) is a .NET component that is used to generate the user interface of an ASP.NET Web application. Web custom controls provide an abstraction and reusability mechanism for web apps. It is implemented as a managed class deriving directly or indirectly from the System.Web.UI.Control base class. (Kothrari, 2003)

f) Client Script

Most of ASP.NET UI Component emits JavaScript, Standard covers the way JavaScript is structured and how component can expose its functionality to the client. This section in at some points crosses with Cross-Browser section.

g) Localization & globalization

Globalization is the process of designing and developing applications that function for multiple cultures, and localization is the process of customizing your application for a given culture and locale as suggested by Microsoft (2006). Proper globalization and localization strategy will allow component to be accessed by wider range of developers and give them tools to use component in multicultural applications.

h) Deployment

Standard rules for deployment section are defining requirements for component assemblies. This section covers issues like side-by-side installation, signing assemblies, licensing, obfuscation and many others.

The author of the Standard attempted to locate as many references and source documents as possible. However, some important rules or recommendation may not have been created, and some applicable references may have been missed. Readers who are aware of an original reference pertaining to an existing rule or recommendation, or who have a suggestion for a new rules, recommendations or sections should submit a feedback on <http://www.webcomponentstandard.org/feedback.aspx>.

3. Conformance issues

Conformance is of interest to the following audiences:

- Those designing, implementing, or maintaining ASP.NET UI components.
- Governmental or commercial entities wishing to procure ASP.NET UI components.
- Educators wishing to teach about ASP.NET UI component development.

- Authors wanting to write about ASP.NET UI components.

As such, conformance is most important. The text in this Standard that specifies requirements is considered normative. Unless stated otherwise, all text is normative. A conforming implementation of ASP.NET UI Component Standard would be accompanied by a document that defines all implementation defined by rules.

The major problem that implementers of the Standard will face is that document is that the rules/recommendation, although good, are mostly unsuited for easy automatic verification. This can be overcome by using recommended validation tools like FxCop or W3C markup validator and by learning process which will improve efficiency of Standard usage.

4. Conclusions and Future Work

In defining this Standard we are following the tendency in the software community to make the development process more formalized and controlled. Compliance with a Standard would allow for the production of high quality components and, as a result, vendors can gain an advantage over their competitors in the software market.

This Standard is intended to be used by implementers, academics, and application programmers. The educational benefits for those who read the Standard will be large. Students and newcomers to the field will profit from the good survey of issues that reminds them of the many facets of Web Component design. Standard can be also used by project managers to be able to control process of development in more organized way.

The development of this Standard started in October 2005. It is expected that there will be future revisions to this Standard, primarily to add new functionality or improve existing rules.

5. References

- Abrams B., Cwalina K. (2005) *Framework Design Guidelines : Conventions, Idioms, and Patterns for Reusable .NET Libraries* (pp. 67 – 103). NY: Addison-Wesley Professional.
- Balley R., Koyani S., Nall J. (2003) *Research-Based Web Design & Usability Guidelines*. National Cancer Institute. Retrieved March 2006 from http://usability.gov/pdfs/guidelines_book.pdf.
- Caldwell, B., Chisholm, W., Slatin J., Vanderheiden G. (2005) *Web Content Accessibility Guidelines 2.0, Working Draft*. Retrieved March 2006, from <http://www.w3.org/TR/WCAG20/>.
- Cameron, R., Michalik, D. (2003) *Building ASP.NET Server Controls*, Apress.
- Clarke, S. (2004) Measuring API Usability. *Dr. Dobbs's Journal Windows/.NET Supplement*, 5, S7-S9. (<http://www.gotdotnet.com/team/brada/APIUsability.pdf>)
- Clary, B. (2003) *Browser Detection and Cross Browser Support*. Retrieved February 2006 from http://developer.mozilla.org/en/docs/Browser_Detection_and_Cross_Browser_Support.
- Doomen, D., Hartog V. (2003) *Coding Standard: C#*. Philips Electronics. Retrieved February 2006, from <http://www.tiobe.com/standards/gemresharpcs.pdf>.
- Duthie, A. (2004) *Adding Design-Time Support to ASP.NET Controls*. Retrieved June 2006 from <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspp/html/ASPNet-AddDesignTimeSupport.asp>

Esposito D. (2006) *Programming Microsoft ASP.NET 2.0 Applications: Advanced Topics*, Microsoft Press.

IEEE (2001). *Draft Standard for Software User Documentation*, IEEEStd1063/ D5.1. 2001. New York: Institute of Electrical and Electronics Engineers.

Koechley, N. (2006) *Graded Browser Support*. Retrieved June 2006, from <http://developer.yahoo.com/yui/articles/gbs/gbs.html>.

Kothari, N., Datye, V. (2003) *Developing Microsoft ASP.NET Server Controls and Components*, Microsoft Press.

Löwy J. (2005) *Programming .NET Components* (pp. 572-587). Sebastopol: O'Reilly Media.

Microsoft (2003) *Language Interoperability Overview*. Retrieved June 2006, from <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpgenref/html/cpconRelationshipToCommonTypeSystemCommonLanguageSpecification.asp>.

Microsoft (2006) *ASP.NET Globalization and Localization*. Retrieved May 2006 from <http://msdn2.microsoft.com/en-us/library/c6zyy3s9.aspx>

Muhamad, F., Miller, M. (2003) *Real World ASP.NET Best Practices*, APress

Walther, S. (2005) *Building ASP.NET 2.0 Web Sites Using Web Standards*. Retrieved February 2006, from <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspp/html/aspnetusstan.asp>