



CSS Namespaces Module Level 3

W3C Recommendation 29 September 2011, edited in place 20 March 2014

This version:

<http://www.w3.org/TR/2014/REC-css-namespaces-3-20140320/>

Latest version:

<http://www.w3.org/TR/css-namespaces-3/>

Editor's Draft:

<http://dev.w3.org/csswg/css-namespaces/>

Previous version:

<http://www.w3.org/TR/2011/REC-css3-namespace-20110929/>

Feedback:

www-style@w3.org with subject line “[css-namespaces] ... *message topic* ...”(archives)

Test Suite:

<http://test.csswg.org/suites/css3-namespace/20090210/>

Editors:

Elika J. Etemad (Invited Expert)

Former Editors:

Anne van Kesteren (Opera Software ASA)

Peter Linss (Netscape Communications)

Chris Lilley (W3C)

Please check the **errata** for any errors or issues reported since publication. See also **translations**.

Copyright © 2014 W3C® (MIT, ERCIM, Keio, Beihang), All Rights Reserved. W3C liability, trademark and document use rules apply.

Abstract

This CSS Namespaces module defines the syntax for using namespaces in CSS. It defines the '@namespace' rule for declaring the default namespace and binding namespaces to namespace prefixes, and it also defines a syntax that other specifications can adopt for using those prefixes in namespace-qualified names. CSS is a language for describing the rendering of structured documents (such as HTML and XML) on screen, on paper, in speech, etc.

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at <http://www.w3.org/TR/>.

This document was produced by the CSS Working Group as a Recommendation.

This document has been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and is endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

W3C encourages everybody to implement this specification. Comments may be sent to the (archived) public mailing list www-style@w3.org (see instructions). When sending e-mail, please put the text "css-namespaces-3" in the subject, preferably like this: "[css-namespaces-3] ...summary of comment..."

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent.

For details of the changes since the previous version, see the Changes section.

Table of Contents

2	Conformance
2.1	Terminology
3	Declaring namespaces: the ‘@namespace’ rule
3.1	Syntax
3.2	Scope
3.3	Declaring Prefixes
4	CSS Qualified Names

Changes

Acknowledgments

References

Normative References

Informative References

Index

Property index

§ 1 Introduction

This section is non-normative.

This CSS Namespaces module defines syntax for using namespaces in CSS. It defines the ‘@namespace’ rule for declaring a default namespace and for binding namespaces to namespace prefixes. It also defines a syntax for using those prefixes to represent namespace-qualified names. It does not define where such names are valid or what they mean: that depends on their context and is defined by a host language, such as Selectors ([SELECT]), that references the syntax defined in the CSS Namespaces module.

Note that a CSS client that does not support this module will (if it properly conforms to CSS’s forward-compatible parsing rules) ignore all ‘@namespace’ rules, as well as all style rules that make use of namespace qualified names. The syntax of delimiting namespace prefixes in CSS was deliberately chosen so that these CSS clients would ignore the style rules rather than possibly match them incorrectly.

§ 2 Conformance

A document or implementation cannot conform to CSS Namespaces alone, but can claim conformance to CSS Namespaces if it satisfies the conformance requirements in this specification when implementing CSS or another host language that normatively references this specification.

Conformance to CSS Namespaces is defined for two classes:

style sheet

A CSS style sheet (or a complete unit of another host language that normatively references CSS Namespaces).

interpreter

Someone or something that interprets the semantics of a style sheet. (CSS user agents fall under this category.)

The conformance requirements are expressed with a combination of descriptive assertions and RFC 2119 terminology. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in the normative parts of this document are to be interpreted as described in RFC 2119. However, for readability, these words do not appear in all uppercase letters in this specification. All of the text of this specification is normative except sections explicitly marked as non-normative, examples, and notes. [RFC2119]

Examples in this specification are introduced with the words "for example" or are set apart from the normative text with `class="example"`, like this:

EXAMPLE 1

This is an example of an informative example.

Informative notes begin with the word "Note" and are set apart from the normative text with `class="note"`, like this:

Note, this is an informative note.

§ 2.1 Terminology

Besides terms introduced by this specification, CSS Namespaces uses the terminology defined in Namespaces in XML 1.0. [XML-NAMES] However, the syntax defined here is not restricted to representing XML element and attribute names and may represent other kinds of namespaces as defined by the host language.

In CSS Namespaces a namespace name consisting of the empty string is taken to represent the null namespace or lack of a namespace.

EXAMPLE 2

For example, given the namespace declarations:

```
@namespace empty "";  
@namespace "";
```

The type selectors `elem`, `|elem`, and `empty|elem` are equivalent.

§ 3 Declaring namespaces: the '@namespace' rule

The '@namespace' at-rule declares a namespace prefix and associates it with a given namespace name (a string). This namespace prefix can then be used in namespace-qualified names such as the CSS qualified names defined below.

EXAMPLE 3

```
@namespace "http://www.w3.org/1999/xhtml";  
@namespace svg "http://www.w3.org/2000/svg";
```

The first rule declares a default namespace `http://www.w3.org/1999/xhtml` to be applied to names that have no explicit namespace component.

The second rule declares a namespace prefix `svg` that is used to apply the namespace `http://www.w3.org/2000/svg` where the `svg` namespace prefix is used.

In CSS Namespaces, as in Namespaces in XML 1.0, the prefix is merely a syntactic construct; it is the **expanded name** (the tuple of local name and namespace name) that is significant. Thus the actual prefixes used in a CSS style sheet, and whether they are defaulted or not, are independent of the namespace prefixes used in the markup and whether these are defaulted or not.

EXAMPLE 4

For example, given the following XML document:

```
<qml:elem xmlns:qml="http://example.com/q-markup"></qml:elem>
```

and the following '@namespace' declarations at the beginning of a CSS file:

```
@namespace Q "http://example.com/q-markup";  
@namespace lq "http://example.com/q-markup";
```

The selectors 'Q|elem' and 'lq|elem' in that CSS file would both match the element <qml:elem>.

(The selector 'qml|elem' would be invalid, because CSS namespaces only recognize prefixes declared in CSS, not those declared by the document language.)

§ 3.1 Syntax

The syntax for the '@namespace' rule is as follows (using the notation from the Grammar appendix of CSS 2.1 [CSS21]):

```
namespace  
  : NAMESPACE_SYM S* [namespace_prefix S*]? [STRING|URI] S* ';' S*  
  ;  
namespace_prefix  
  : IDENT  
  ;
```

with the new token:

```
@{N}{A}{M}{E}{S}{P}{A}{C}{E} {return NAMESPACE_SYM;}
```

Any '@namespace' rules must follow all @charset and @import rules and precede all other non-ignored at-rules and style rules in a style sheet. For CSS syntax this adds [namespace [S|CDO|CDC]*]* immediately after [import [S|CDO|CDC]*]* in the stylesheet grammar.

A syntactically invalid '@namespace' rule (whether malformed or misplaced) must be ignored. A CSS style sheet containing an invalid '@namespace' rule is not a valid style sheet.

A URI string parsed from the URI syntax must be treated as a literal string: as with the STRING syntax, no URI-specific normalization is applied.

All strings—including the empty string and strings representing invalid URIs—are valid namespace names in '@namespace' declarations.

§ 3.2 Scope

The namespace prefix is declared only within the style sheet in which its '@namespace' rule appears. It is not declared in any style sheets importing or imported by that style sheet, nor in any other style sheets applying to the document.

§ 3.3 Declaring Prefixes

A **namespace prefix**, once declared, represents the namespace for which it was declared and can be used to indicate the namespace of a namespace-qualified name. Namespace prefixes are, like CSS counter names, case-sensitive.

If in the namespace declaration the namespace prefix is omitted, then the namespace so declared is the default namespace. The **default namespace** may apply to names that have no explicit namespace prefix: modules that employ namespace prefixes must define in which contexts the default namespace applies. For example, following [XML-NAMES], in Selectors [SELECT] the default namespace applies to type selectors—but it does not apply to attribute selectors. There is no default value for the default namespace: modules that assign unqualified names to the default namespace must define how those unqualified names are to be interpreted when no default namespace is declared.

Note: Note that using default namespaces in conjunction with type selectors can cause UAs that support default namespaces and UAs that don't support default namespaces to interpret selectors differently.

If a namespace prefix or default namespace is declared more than once only the last declaration shall be used. Declaring a namespace prefix or default namespace more than once is nonconforming.

§ 4 CSS Qualified Names

A **CSS qualified name** is a name explicitly located within (associated with) a namespace. To form a qualified name in CSS syntax, a namespace prefix that has been declared within scope is prepended to a local name (such as an element or attribute name), separated by a "vertical bar"(|, U+007C). The prefix, representing the namespace for which it has been declared, indicates the namespace of the local name. The prefix of a qualified name may be omitted to indicate that the name belongs to no namespace, i.e. that the namespace name part of the expanded name has no value. Some contexts (as defined by the host language) may allow the use of an asterisk (*, U+002A) as a wildcard prefix to indicate a name in any namespace, including no namespace.

EXAMPLE 5

Given the namespace declarations:

```
@namespace toto "http://toto.example.org";  
@namespace "http://example.com/foo";
```

In a context where the default namespace applies

toto|A

represents the name A in the `http://toto.example.org` namespace.

|B

represents the name B that belongs to no namespace.

***|C**

represents the name C in any namespace, including no namespace.

D

represents the name D in the `http://example.com/foo` namespace.

The syntax for the portion of a CSS qualified name before the local name is given below, both for qualified names that allow wildcard prefixes (wqname) and for qualified names that disallow wildcard prefixes (qname). (The syntax uses notation from the Grammar appendix of CSS 2.1. [CSS21] Note this means that comments, but not white space, are implicitly allowed between tokens.):


```

qname_prefix
  : namespace_prefix? '|'
  ;
wqname_prefix
  : [ namespace_prefix? | '*' ] '|'
  ;
qname
  : qname_prefix? ident
  ;
wqname
  : wqname_prefix? ident
  ;
wqwnname
  : wqname_prefix? [ ident | '*' ]
  ;

```

CSS qualified names can be used in (for example) selectors and property values as described in other modules. Those modules must define handling of namespace prefixes that have not been properly declared. Such handling should treat undeclared namespace prefixes as a parsing error that will cause the selector or declaration (etc.) to be considered invalid and, in CSS, ignored.

EXAMPLE 6

For example, the Selectors module [SELECT] defines a type selector with an undeclared namespace prefix to be an invalid selector, and CSS [CSS21] requires style rules with an invalid selector to be completely ignored.

§ Changes

Changes made since the 29 September 2011 Recommendation:

- Added predefined ‘[qname](#)’, ‘[wqname](#)’, and ‘[wqwnname](#)’ productions, to make those constructs easier for other specs to use.

§ Acknowledgments

This draft borrows heavily from earlier drafts on CSS namespace support by Chris Lilley and by Peter Linss and early (unpublished) drafts on CSS and XML by Håkon Lie and Bert Bos, and XML Namespaces and CSS by Bert Bos and Steven Pemberton. Many current and former members of the CSS Working Group have contributed to this document. Discussions on www-style@w3.org and in other places have also contributed ideas to this specification. Special thanks goes to L. David Baron, Karl Dubost, Ian Hickson, Björn Höhrmann, and Lachlan Hunt for their comments.

§ References

§ Normative References

¶ [CSS21]

Bert Bos; et al. Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. 7 June 2011. W3C Recommendation. URL: <http://www.w3.org/TR/2011/REC-CSS2-20110607>

¶ [RFC2119]

S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. URL: <http://www.ietf.org/rfc/rfc2119.txt>

¶ [XML-NAMES]

Tim Bray; et al. Namespaces in XML 1.0 (Third Edition). 8 December 2009. W3C Recommendation. URL: <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

§ Informative References

¶ [SELECT]

Tantek Çelik; et al. Selectors Level 3. 29 September 2011. W3C Recommendation. URL: <http://www.w3.org/TR/2011/REC-css3-selectors-20110929/>

§ Index

CSS qualified name, **4**

default namespace, **3.3**

expanded name, **3**

interpreter, **2**

namespace prefix, **3.3**

style sheet, **2**

§ Property index

No properties defined.