[MS-PICSRL]:
Internet Explorer PICSRules Standards Support Document

Revision Summary

Date	Revision History	Revision Class	Comments
03/17/2010	0.1	New	Released new document.
03/26/2010	1.0	None	Introduced no new technical or language changes.
05/26/2010	1.2	None	Introduced no new technical or language changes.
09/08/2010	1.3	Major	Significantly changed the technical content.
02/10/2011	2.0	No change	Introduced no new technical or language changes.
02/22/2012	3.0	Major	Significantly changed the technical content.
07/25/2012	3.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction

This document describes the level of support provided by Windows® Internet Explorer® for the *PICSRules 1.1* [W3C-PICS-Rules], W3C Recommendation 29 December 1997, revised on 24 November 2009. Internet Explorer displays webpages written in HTML.

The [W3C-PICS-Rules] specification may contain guidance for authors of webpages and browser users, in addition to user agents (browser applications). Statements found in this document apply only to normative requirements in the specification targeted to user agents, not those targeted to authors.

1.1 Glossary

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[W3C-PICS-Rules] Evans, C., Feather, Clive D. W., Hopmann, A., Presler-Marshall, M., and Resnick, P., "PICSRules 1.1", W3C Recommendation 29 Dec 1997 (revised 24-Nov-2009), http://www.w3.org/TR/REC-PICSRules/

1.2.2 Informative References

None.

1.3 Microsoft Implementations

The following Microsoft products implement some portion of [W3C-PICS-Rules]:

- Windows® Internet Explorer® 7
- Windows® Internet Explorer® 8
- Windows® Internet Explorer® 9
- Windows® Internet Explorer® 10

In addition, each version of Windows® Internet Explorer® implements multiple document modes, which can vary individually in their support of the standard. The following table lists the document modes available in each version of Internet Explorer:

Browser Version	Documents Modes Supported
Internet Explorer 7	Quirks Mode Standards Mode
Internet Explorer 8	Quirks Mode IE7 Mode IE8 Mode
Internet Explorer 9	Quirks Mode IE7 Mode IE8 Mode IE9 Mode
Internet Explorer 10	Quirks Mode IE7 Mode IE8 Mode IE9 Mode IE10 Mode

Throughout this document, the document mode appears first followed by the browser version in parentheses. Only those document modes and versions of Internet Explorer for which there is a variation note will be listed. If the document mode is not listed, conformance to the specification can be assumed.

Note "Standards mode" in Internet Explorer 7 and "IE7 mode" in Internet Explorer 8 refer to the same document mode. "IE7 mode" is the preferred way of referring to this document mode across all versions of the browser.

1.4 Standards Support Requirements

To conform to [W3C-PICS-Rules] a user agent must implement all required portions of the specification. Any optional portions that have been implemented must also be implemented as described by the specification. Normative language is usually used to define both required and optional portions. (For more information, see [RFC2119].)

The following table lists the sections of [W3C-PICS-Rules] and whether they are considered normative or informative.

Sections	Normative/Informative
Abstract Introduction Definitions The PICSRules language: examples Full syntax	Informative
General Semantics Control Flow	Normative

Sections	Normative/Informative
Extension mechanism	

1.5 Notation

The following notations are used in this document to differentiate between notes of clarification, variation from the specification, and extension points.

Notation	Explanation
C####	This identifies a clarification of ambiguity in the target specification. This includes imprecise statements, omitted information, discrepancies, and errata. This does not include data formatting clarifications.
V####	This identifies an intended point of variability in the target specification such as the use of MAY, SHOULD, or RECOMMENDED. (See [RFC2119].) This does not include extensibility points.
E####	Because the use of extensibility points (such as optional implementation-specific data) can impair interoperability, this profile identifies such points in the target specification.

For document mode and browser version notation, see also section 1.3.

2 Standards Support Statements

This section contains a full list of variations, clarifications, and extension points in the Microsoft implementation of [W3C-PICS-Rules].

- Section 2.1 includes only those variations that violate a MUST requirement in the target specification.
- Section 2.2 describes further variations from MAY and SHOULD requirements.
- Section 2.3 identifies variations in error handling.
- Section 2.4 identifies variations that impact security.

2.1 Normative Variations

There are no variations from [W3C-PICS-Rules].

2.2 Clarifications

The following subsections identify clarifications to recommendations made by [W3C-PICS-Rules].

2.2.1 [W3C-PICS-Rules] Full syntax

V0001:

The specification states:

```
Basic structure
PICSRules rules are based on a limited form of an S-expression, namely a
parenthesized attribute-value pair. A value is either a quoted string or a
parenthesized list of additional attribute-value pairs, thus allowing nesting. When
a value for an attribute is a list of further pairs, there is a concept known as a
"primary attribute". The name of the primary attribute may be omitted, for the sake
of readability, so that only the value of the primary attribute is specified. A
parser can syntactically distinguish values from attributes (values begin with
either a quote or left parenthesis); any values that are not paired with attribute
names automatically belong to the primary attribute. When a value for an attribute
is a list of pairs, the list MUST include the primary attribute-value pair (with or
without the primary attribute name specified); it MAY contain additional attribute-
value pairs. The general grammar for these limited S-expressions is:
attrvalpair:: attribute whitespace value | value

attribute:: alphanumstr

value:: quotedstring |'(' attrvalpair+ ')'

quotedstring:: '"'notdoublequotechar*'"' | "'"notsinglequotechar*"'"

alphanumstr:: (alphanum | '.')+

whitespace:: ' ' | '\t' | '\r' | '\n'

alphanum:: '0' - '9' | 'A' - 'Z' | 'a' - 'z'

notdoublequotechar :: any Unicode character except "
```

```
notsinglequotechar :: any Unicode character except '
```

All Document Modes (All Versions)

Attribute-value pairs that do not have white spaces to separate the attribute and value are supported.

V0002:

The specification states:

```
The other quoting character may appear within a string. In order to accommodate the
use of both single and double quotes inside strings, the following escaping
conventions apply:
" may be encoded as %22
' may be encoded as %27
% may be encoded as %25
% followed by anything other than 22, 27, or 25 is syntactically invalid
Note that, although ", ', and % are encoded using the % hex hex encoding rule used
for special characters in URLs, other % hex hex combinations are not valid and are
not considered encodings of other characters.
```

All Document Modes (All Versions)

All escape characters (percent sign (%) followed by any character) are valid, including % followed by 22, 27, or 25.

V0003:

The specification states:

```
Comments
The PICSRules syntax, which will be presented below, has a facility for descriptive
text which can be shown to a user, in addition to various statements which
influence the behavior of user-agents. However, it is frequently useful to have
"source-level" comments - comments which are intended to individuals writing and/or
editing rules, but which are not intended for display to end users. This is
analogous to placing comments in source code; in an effort to encourage rule
authors to write clear rules, we provide a facility for placing comments into
PICSRules rules.
The syntax of a comment is:
comment:: '{' comment-text* '}'

comment-text:: any characters except '}'
Note that a result of the above syntax is that comments may not be nested.
Comments may appear anywhere in PICSRules rules. A user-agent MAY remove the
comments during lexical analysis of the rule; text within comments MUST NOT
influence the interpretation of the rule in any manner. Note also that user-agents
which generate or export PICSRules rules MAY choose to strip out comments before
generating, exporting, or transmitting them.
```

All Document Modes (All Versions)

Comments in PICSRules rules are not supported.

V0004:

The specification states:

```
An application program will invoke a rule evaluator, providing a rule and a URL,
and perhaps labels that were embedded in the document associated with the URL or
passed in the HTTP headers along with the document associated with the URL. A yes
(accept) or no (reject) answer is returned. The rule evaluator SHOULD also return
the explanation string associated with the policy clause that determines the final
answer, if such an explanation string is provided.
```

All Document Modes (All Versions)

The rule evaluator does not return an explanation string associated with the policy clause that determines the final answer, if such an explanation string is provided.

V0005:

The specification states:

```
source
This clause gives information about where the rule came from. There are 4
attributes defined for source: sourceURL, creationTool, author, and lastModified.
The primary attribute is sourceURL.
The sourceURL attribute gives the "rule's URL". It provides a location where a
human user of this rule can go to get more information about the rule and/or its
creator. The value of this attribute should be a URL here a user can find a human-
readable description of this rule.
The creationTool attribute gives the ability to identify the tool, if any, that was
used to create this rule. This is analogous to the User-Agent string in HTTP. The
value of the creationTool is a quoted string. The string should be in the format
toolname/version, as in "Cool-PICS-Rule-Editor/1.04".
The author attribute gives the e-mail address of the individual or organization who
produced this rule. The value associated with this attribute MUST be a quoted e-
mail address.
The lastModified attribute gives the date and time that this rule was last
modified. The value MUST be a quoted-ISO-date, as defined in the PICS-1.1 Label
Syntax and Communication Protocols.
```

All Document Modes (All Versions)

creationTool values that do not conform to the toolname/version format and **author** values that do not conform to the quoted e-mail format are acceptable.

V0006:

The specification states:

```
serviceinfo
The bureauUnavailable attribute specifies what to do when none of the label
bureau(s) listed in bureauURL attributes can be contacted. The defined values for
this attribute are "PASS" and "FAIL", which cause the rule to return the
corresponding value, regardless of what other labels are found.
The ratfile attribute presents the machine-readable rating system description (also
know as "RAT file") that is used by this rating service. This may be specified in
one of two ways: the value may be a quoted string which contains the entire
```

```
machine-readable service description, or it may be of the syntax "[RATFile-URL]",
where RATFile-URL is the URL of the rating system description; a user-agent SHOULD
assume that dereferencing this URL will produce a document of type
application/pics-service. There is no default value for the ratfile attribute. If
the quoted string contains the machine-readable service description, then it MUST
be escaped as mentioned above.
```

All Document Modes (All Versions)

Neither the **bureauUnavailable** attribute nor the **ratfile** attribute are supported.

V0007:

The specification states:

```
opt-extension-clause
opt-extension-clause and req-extension-clause are the extension mechanisms in
PICSRules; they are modeled after the extension mechanism in the PICS label format.
More information on the extension mechanism is given below.
The opt-extension-clause has two defined attributes: extension-name and shortname.
The value of the extension-name attribute specifies the name of an extension that
will be used by this rule. The name of the extension is the quotedURL; this URL
should point to a human-readable description of this extension. URLs are used for
extension names to ensure uniqueness without requiring a central naming body. The
value of the shortname attribute is a quoted string, but MUST use only valid
attribute name characters (a-z, A-Z, 0-9). The shortname is used as a prefix of
attribute names, to identify attributes defined by this extension.
If a user-agent receives a rule which contains an optextension which it does not
recognize, the user-agent should process the rule, ignoring any clauses it does not
recognize. This means that any optional extensions MUST use the attribute-value
syntax given above, so as to not break existing parsers. Note that declaring the
use of an optional extension may appear to be redundant, as unrecognized attribute-
value pairs are discarded by user-agents. The purpose of the optextension construct
is for use as a documentation mechanism. User-agents MAY also display the names of
optional extensions used by a rule, asking the user for confirmation, before making
use of a rule.
```

All Document Modes (All Versions)

The following variations apply:

- The **shortname** attribute can use any characters, not just (a–z, A–Z, 0–9). Therefore, the exact attribute-value syntax for optional extensions is not used.
- The names of any optional extensions used by a rule are not displayed.

V0008:

The specification states:

```
req-extension-clause
This clause has two defined attributes: extension-name and shortname. The value of
the extension-name attribute specifies the name of an extension that will be used
by this rule. The name of the extension is the quotedURL; this URL should point to
a human-readable description of this extension. URLs are used for extension names
to insure uniqueness without requiring a central naming body. The value of the
```

```
shortname attribute is a quoted string, but MUST use only valid attribute name
characters (a-z, A-Z, 0-9). The shortname is used as a prefix of attribute names,
to identify attributes defined by this extension.
If a user-agent is asked to process a request about the acceptability of a URL,
using a rule which contains a req-extension-clause which the user agent does not
recognize, the user agent should signal an error.
```

All Document Modes (All Versions)

The **shortname** attribute can use any characters, not just (a–z, A–Z, 0–9). Therefore, the exact attribute-value syntax for required extensions is not used.

V0009:

The specification states:

```
Restrictions
The following semantic restrictions are imposed on rules:
The name, and source clauses MUST NOT appear more than once each in a PICSRules
rule.
The optextension, reqextension, and serviceinfo clauses MAY appear more than once
in a PICSRules rule.
Each Policy clause MUST have exactly one attribute from the set of {AcceptIf,
RejectIf, AcceptUnless, RejectUnless, AcceptByURL, RejectByURL}.
A rule MAY contain any number of Policy clauses.
A Policy clause MUST NOT contain more than one explanation attribute.
The shortname attribute of an extension clause or a service clause takes a quoted
string as a value, but that string MUST include only characters that are acceptable
for use in attribute names.
A PICSRules parser MUST maintain the order of the values (or value-lists) given for
the attributes in a rule.
```

All Document Modes (All Versions)

The following variations apply:

- Policy clauses that contain multiple attributes are acceptable.
- The **shortname** attribute can use any characters, not just (a–z, A–Z, 0–9).

V0010:

The specification states:

```
ipwild :: ipcomponent '.' ipcomponent '.' ipcomponent '.' ipcomponent
ipcomponent :: integer between '0' and '255' inclusive
bitlength :: integer between '0' and '32' inclusive
```

All Document Modes (All Versions)

The following variations apply:

- Values greater than 255 and less than 0 are valid for **ipcomponent**.
- Values greater than 32 and less than 0 are valid for **bitlength**.

V0011:

The specification states:

```
When comparing a URLpattern to a URL, the rule interpreter MUST NOT unencode the
URL (e.g., do not convert %2F  to  /). If the pattern can be interpreted as an
internet-pattern, then the pattern is divided into its component parts and the URL
matches the pattern if a match occurs on every component that is included in the
pattern.
```

All Document Modes (All Versions)

When comparing a **URLpattern** to a URL, the rule interpreter decodes the URL (for example, it converts %2F to /).

V0012:

The specification states:

```
user
'*' at the beginning or end of the pattern matches any number of characters in the
URL string. '%*' at the beginning or end of the pattern matches the single
character '*' in the URL string. Characters in the middle of the pattern must match
exactly the characters in the URL string; this comparison is case-sensitive. A user
component of "*" in the pattern also matches URLs that omit the user component. If
the user component is omitted from the pattern, there is a match only if the user
component is also omitted from the URL.
```

All Document Modes (All Versions)

User components in the URL are not supported.

V0013:

The specification states:

```
host
'*' at the beginning of the pattern matches any number of characters in the URL
string. '%*' at the beginning of the pattern matches the single character '*' in
the URL string. Subsequent characters in the pattern must exactly match the
remaining characters in the URL string; this comparison is case-insensitive.  Note
that if the pattern specifies a host name (or a host name with wildcards), it does
not match a URL that specifies an IP address, even if the host name in the pattern
would resolve to the IP address in the URL. This avoids the need to perform
expensive reverse DNS lookups based on IP addresses in URLs. Either a host or  an
ipwild component MUST be specified in the URL pattern.
```

All Document Modes (All Versions)

Host or ipwild components are not required to be specified in the URL pattern.

2.3 Error Handling

There are no additional considerations for error handling.

2.4 Security

There are no additional security considerations.

3 Change Tracking

This section identifies changes that were made to the [MS-PICSRL] protocol document between the February 2012 and July 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1 Introduction	Updated document to remove beta tagging.	N	Content updated.

4 Index