The Platform for Privacy Preferences 1.0 (P3P1.0) Specification

W3C Recommendation 16 April 2002

This Version:
http://www.w3.org/TR/2002/REC-P3P-20020416/
Latest Version:
http://www.w3.org/TR/P3P/
Previous Version:
http://www.w3.org/TR/2002/PR-P3P-20020128/

Editor:
Massimo Marchiori, W3C / MIT / University of Venice, (massimo@w3.org)
Authors:
Lorrie Cranor, AT&T
Marc Langheinrich, ETH Zurich
Massimo Marchiori, W3C / MIT / University of Venice
Martin Presler-Marshall, IBM
Joseph Reagle, W3C/MIT

Please refer to the **errata** for this document, which may include some normative corrections.

See also translations.

Abstract

This is the specification of the Platform for Privacy Preferences (P3P). This document, along with its normative references, includes all the specification necessary for the implementation of interoperable P3P applications.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.

This is the W3C Recommendation of the the Platform for Privacy Preferences 1.0 (P3P1.0) Specification.

This document has been reviewed by W3C Members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference 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.

This document has been produced by the P3P Specification Working Group as part of the Privacy Activity in the W3C Technology & Society Domain.

Patent disclosures relevant to this specification may be found on the P3P1.0 patent disclosure page, in conformance with W3C policy.

Please report errors in this document to www-p3p-public-comments@w3.org (publicly archived).

The list of known errors in this specification is available at http://www.w3.org/2002/04/P3Pv1-errata.

The English version of this specification is the only normative version. Information about translations of this document (if any) is available at http://www.w3.org/2002/04/P3Pv1-translations.

A list of current public W3C Technical Reports can be found at http://www.w3.org/TR.

Table of Contents

1. Introduction

The Platform for Privacy Preferences Project (P3P) enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. P3P user agents will allow users to be informed of site practices (in both machine- and human-readable formats) and to automate decision-making based on these practices when appropriate. Thus users need not read the privacy policies at every site they visit.

Although P3P provides a technical mechanism for ensuring that users can be informed about privacy policies before they release personal information, it does not provide a technical mechanism for making sure sites act according to their policies. Products implementing this specification MAY provide some assistance in that regard, but that is up to specific implementations and outside the scope of this specification. However, P3P is complementary to laws and self-regulatory programs that can provide enforcement mechanisms. In addition, P3P does not include mechanisms for transferring data or for securing personal data in transit or storage. P3P may be built into tools designed to facilitate data transfer. These tools should include appropriate security safeguards.

1.1 The P3P1.0 Specification

The P3P1.0 specification defines the syntax and semantics of P3P privacy policies, and the mechanisms for associating policies with Web resources. P3P policies consist of statements made using the P3P *vocabulary* for expressing privacy practices. P3P policies also reference elements of the P3P *base data schema* -- a standard set of data elements that all P3P user agents should be aware of. The P3P specification includes a mechanism for defining new data elements and data sets, and a simple mechanism that allows for extensions to the P3P vocabulary.

1.1.1 Goals and Capabilities of P3P1.0

P3P version 1.0 is a protocol designed to inform Web users of the data-collection practices of Web sites. It provides a way for a Web site to encode its data-collection and data-use practices in a machine-readable XML format known as a *P3P policy*. The P3P specification defines:

- A standard schema for data a Web site may wish to collect, known as the "P3P base data schema"
- A standard set of uses, recipients, data categories, and other privacy disclosures
- An XML format for expressing a privacy policy
- A means of associating privacy policies with Web pages or sites, and cookies
- A mechanism for transporting P3P policies over HTTP

The goal of P3P version 1.0 is twofold. First, it allows Web sites to present their data-collection practices in a standardized, machine-readable, easy-to-locate manner. Second, it enables Web users to understand what data will be collected by sites they visit, how that data will be used, and what data/uses they may "opt-out" of or "opt-in" to.

1.1.2 Example of P3P in Use

As an introduction to P3P, let us consider one common scenario that makes use of P3P. Claudia has decided to check out a store called CatalogExample, located at http://www.catalog.example.com/. Let us assume that CatalogExample has placed P3P policies on all their pages, and that Claudia is using a Web browser with P3P built in.

Claudia types the address for CatalogExample into her Web browser. Her browser is able to automatically fetch the P3P policy for that page. The policy states that the only data the site collects on its home page is the data found in standard HTTP access logs. Now Claudia's Web browser checks this policy against the preferences Claudia has given it. Is this policy acceptable to her, or should she be notified? Let's assume that Claudia has told her browser that this is acceptable. In this case, the homepage is displayed normally, with no pop-up messages appearing. Perhaps her browser displays a small icon somewhere along the edge of its window to tell her that a privacy policy was given by the site, and that it matched her preferences.

Next, Claudia clicks on a link to the site's online catalog. The catalog section of the site has some more complex software behind it. This software uses cookies to implement a "shopping cart" feature. Since more information is being gathered in this section of the Web site, the Web server provides a separate P3P policy to cover this section of the site. Again, let's assume that this policy matches Claudia's preferences, so she gets no pop-up messages. Claudia continues and selects a few items she wishes to purchase. Then she proceeds to the checkout page.

The checkout page of CatalogExample requires some additional information: Claudia's name, address, credit card number, and telephone number. Another P3P policy is available that describes the data that is collected here and states that her data will be used only for completing the current transaction, her order.

Claudia's browser examines this P3P policy. Imagine that Claudia has told her browser that she wants to be warned whenever a site asks for her telephone number. In this case, the browser will pop up a message saying that this Web site is asking for her telephone number, and explaining the contents of the P3P statement. Claudia can then decide if this is acceptable to her. If it is acceptable, she can continue with her order; otherwise she can cancel the transaction.

Alternatively, Claudia could have told her browser that she wanted to be warned only if a site is asking for her telephone number and was going to give it to third parties and/or use it for uses other than completing the current transaction. In that case, she would have received no prompts from her browser at all, and she could proceed with completing her order.

Note that this scenario describes one hypothetical implementation of P3P. Other types of user interfaces are also possible.

1.1.3 P3P Policies

P3P policies use an XML with namespaces (cf. [XML] and [XML-Name]) encoding of the P3P vocabulary to provide contact information for the legal entity making the representation of privacy practices in a policy, enumerate the types of data or data elements collected, and explain how the data will be used. In addition, policies identify the data recipients, and make a variety of other disclosures including information about dispute resolution, and the address of a site's human-readable privacy policy. P3P policies must cover all relevant data elements and practices. However, legal issues regarding law enforcement demands for information are not addressed by this specification. It is possible that a site that otherwise abides by its policy of not redistributing data to others may be required to do so by force of law. P3P declarations are positive, meaning that sites state what they do, rather than what they do not do. The P3P vocabulary is designed to be descriptive of a site's practices rather than simply an indicator of compliance with a particular law or code of conduct. However, user agents may be developed that can test whether a site's practices are compliant with a law or code.

P3P policies represent the practices of the site. Intermediaries such as telecommunication providers, Internet service providers, proxies and others may be privy to the exchange of data between a site and a user, but their practices may not be governed by the site's policies. In addition, note that each P3P policy is applied to specific Web resources (Web pages, images, cookies, etc.) listed in a policy reference file. By placing one or more P3P policies on a Web site, a company or organization does not make any statements about the privacy practices associated with other Web resources not mentioned in their policy reference file, with other online activities that do not involve data collected on Web sites covered by their P3P policy, or with offline activities that do not involve data collected on Web sites covered by their P3P policy.

In cases where the P3P vocabulary is not precise enough to describe a Web site's practices, sites should use the vocabulary terms that most closely match their practices and provide further explanations (as stated in Section 3.2). However, policies MUST NOT make false or misleading statements.

1.1.4 P3P User Agents

P3P1.0 user agents can be built into Web browsers, browser plug-ins, or proxy servers. They can also be implemented as Java applets or JavaScript; or built into electronic wallets, automatic form-fillers, or other user data management tools. P3P user agents look for references to a P3P policy at a well-known location, in P3P headers in HTTP responses, and in P3P `link` tags embedded in HTML content. These references indicate the location of a relevant P3P policy. User agents can fetch the policy from the indicated location, parse it, and display symbols, play sounds, or generate user prompts that reflect a site's P3P privacy practices. They can also compare P3P policies with privacy preferences set by the user and take appropriate actions. P3P can perform a sort of "gate keeper" function for data transfer mechanisms such as electronic wallets and automatic form fillers. A P3P user agent integrated into one of these mechanisms would retrieve P3P policies, compare them with user's preferences, and authorize the release of data only if a) the policy is consistent with the user's preferences and b) the requested data transfer is consistent with the policy. If one of these conditions is not met, the user might be informed of the discrepancy and given an opportunity to authorize the data release themselves.

The P3P1.0 Specification places few requirements on the user interfaces of P3P user agents. Thus user agent implementers may each make their own choices about what words and symbols to present to users to provide information about a Web site's privacy policy. Implementers need not use the definitions found in this specification verbatim in their user interfaces. They should, however, make sure that whatever information they present to the user accurately represents the P3P policies described, as per Appendix 7, "P3P Guiding Principles".

1.1.5 Implementing P3P1.0 on Servers

Web sites can implement P3P1.0 on their servers by translating their human-readable privacy policies into P3P syntax and then publishing the resulting files along with a policy reference file that indicates the parts of the site to which the policy applies. Automated tools can assist site operators in performing this translation. P3P1.0 can be implemented on existing HTTP/1.1-compliant Web servers without requiring additional or upgraded software. Servers may publish their policy reference files at a well-known location, or they may reference their P3P policy reference files in HTML/XHTML content using a `link` tag. Alternatively, compatible servers may be configured to insert a P3P extension header into all HTTP responses that indicates the location of a site's P3P policy reference file.

Web sites have some flexibility in how they use P3P: they can opt for one P3P policy for their entire site or they can designate different policies for different parts of their sites. A P3P policy MUST cover all data generated or exchanged as part of a site's HTTP interactions with visitors. In addition, some sites may wish to write policies that cover all data an entity collects, regardless of how the data is collected.

1.1.6 Future Versions of P3P

Significant sections were removed from earlier drafts of the P3P1.0 specification in order to facilitate rapid implementation and deployment of a P3P first step. A future version of the P3P specification might incorporate those features after P3P1.0 is deployed. Such specification would likely include improvements based on feedback from implementation and deployment experience as well as four major components that were part of the original P3P vision but not included in P3P1.0:

- a mechanism to allow sites to offer a choice of P3P policies to visitors
- a mechanism to allow visitors (through their user agents) to explicitly agree to a P3P policy
- mechanisms to allow for non-repudiation of agreements between visitors and Web sites
- a mechanism to allow user agents to transfer user data to services

1.2 About this Specification

This document, along with its normative references, includes all the specification necessary for the implementation of interoperable P3P applications.

The following key words are used throughout the document and have to be read as interoperability requirements. This specification uses words as defined in RFC2119 [KEY] for defining the significance of each particular requirement. These words are:

MUST or MUST NOT
: This word or the adjective "required" means that the item is an absolute requirement of the specification.

SHOULD or SHOULD NOT
: This word or the adjective "rcommended" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

MAY
: This word or the adjective "optional" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

The P3P specification defines, with the exception of section 2.2.2, section 2.2.3 and section 4, an *XML with namespaces* syntax (cf. [XML] and [XML-Name]). In the following, for the sake of brevity we will liberally talk about "XML", meaning the more accurate "XML with namespaces".

A BNF-like notation is also used thorough the specification: the [ABNF] notation used in this specification is specified in RFC2234 and summarized in Appendix 6. However, note that in the case of XML syntax, such ABNF syntax is only a grammar representative used to enhance readability (lacking, for example, all the syntactic flexibilities that are implicitly included in XML, e.g. whitespace rules, quoting using either single quote (') or double quote ("), character escaping, comments, case sensitivity, order of attributes, namespace handling), and as such it has no normative value. All the XML syntax defined in this specification MUST conform to the XML Schema for P3P (see Appendix 4), which, together with the other constraints expressed in this specification using natural language, constitutes the *normative* definition.

The (non-normative) DTD provided in Appendix 5 MAY be used to verify that P3P files are valid. However, there are some valid files that may be rejected if checked against the DTD due to their use of namespaces.

As far as the non-XML syntax defined in this specification is concerned (section 2.2.2 defining P3P's HTTP header, section 2.2.3 defining usage of P3P in HTML, and section 4 defining compact policies), instead, the ABNF notation (together with the other constraints expressed in this specification using natural language) constitutes the *normative* definition.

1.3 Terminology

Character
: Strings consist of a sequence of zero or more characters, where a character is defined as in the XML Recommendation [XML]. A single character in P3P thus corresponds to a single Unicode abstract character with a single corresponding Unicode scalar value (see [UNICODE]).

Data Element
: An individual data entity, such as last name or telephone number. For interoperability, P3P1.0 specifies a base set of data elements.

Data Category
: A significant attribute of a data element or data set that may be used by a trust engine to determine what type of element is under discussion, such as physical contact information. P3P1.0 specifies a set of data categories.

Data Set
: A known grouping of data elements, such as "`user.home-info.postal`". The P3P1.0 base data schema specifies a number of data sets.

Data Schema
: A collection of data elements and sets defined using the P3P1.0 `DATASCHEMA` element. P3P1.0 defines a standard data schema called the *P3P base data schema*.

Data Structure
: A hierarchical description of a set of data elements. A data set can be described according to its data structure. P3P1.0 defines a set of basic datastructures that are used to describe the data sets in the P3P base data schema.

Equable Practice
: A practice that is very similar to another in that the purpose and recipients are the same or more constrained than the original, and the other disclosures are not substantially different. For example, two sites with otherwise similar practices that follow different -- but similar -- sets of industry guidelines.

Identified Data
: Data that reasonably can be used by the data collector to identify an individual.

Policy
: A collection of one or more privacy statements together with information asserting the identity, URI, assurances, and dispute resolution procedures of the service covered by the policy.

Practice
: The set of disclosures regarding data usage, including purpose, recipients, and other disclosures.

Preference
: A rule, or set of rules, that determines what action(s) a user agent will take. A preference might be expressed as a formally defined computable statement (e.g., the [APPEL] preference exchange language).

Purpose
: The reason(s) for data collection and use.

Repository
: A mechanism for storing user information under the control of the user agent.

Resource
: A network data object or service that can be identified by a URI. Resources may be available in multiple representations (e.g. multiple languages, data formats, size, and resolutions) or vary in other ways.

Safe Zone
: Part of a Web site where the service provider performs only minimal data collection, and any data that is collected is used only in ways that would not reasonably identify an individual.

Service
: A program that issues policies and (possibly) data requests. By this definition, a service may be a server (site), a local application, a piece of locally active code, such as an ActiveX control or Java applet, or even another user agent. Typically, however, a service is usually a Web site. In this specification the terms "service" and "Web site" are often used interchangeably.

Service Provider (Data Controller, Legal Entity)
: The person or legal entity which offers information, products or services from a Web site, collects information, and is responsible for the representations made in a practice statement.

Statement
: A P3P statement is a set of privacy practice disclosures relevant to a collection of data elements.

URI

A Uniform Resource Identifier used to locate Web resources. For definitive information on URI syntax and semantics, see [URI]. URIs that appear within XML or HTML have to be treated as specified in [CHARMODEL], section Character Encoding in URI References. This does not apply to URIs appearing in HTTP header fields; the URIs there should always be fully escaped.

User

An individual (or group of individuals acting as a single entity) on whose behalf a service is accessed and for which personal data exists. P3P policies describe the collection and use of personal data about this individual or group.

User Agent

A program whose purpose is to mediate interactions with services on behalf of the user under the user's preferences. A user may have more than one user agent, and agents need not reside on the user's desktop, but *any agent must be controlled by and act on behalf of only the user*. The trust relationship between a user and his or her agent may be governed by constraints outside of P3P. For instance, an agent may be trusted as a part of the user's operating system or Web client, or as a part of the terms and conditions of an ISP or privacy proxy.

2. Referencing Policies

2.1 Overview and Purpose of Policy References

Locating a P3P policy is one of the first steps in the operation of the P3P protocol. Services use policy references to state what policy applies to a specific URI or set of URIs. User agents use policy references to locate the privacy policy that applies to a Web resource, so that they can process that policy for the benefit of their user.

Policy references are used extensively as a performance optimization. P3P policies are typically several kilobytes of data, while a URI that references a privacy policy is typically less than 100 bytes. In addition to the bandwidth savings, policy references also reduce the need for computation: policies can be uniquely associated with URIs, so that a user agent need only parse and process a policy once rather than process it with every document to which the policy applies. Furthermore, by placing the information about relevant policies in a centralized location, Web site administration is simplified.

A policy reference file is used to associate P3P policies with certain regions of URI-space. The policy reference file is an XML with namespaces (see [XML] and [XML-Name]) file that can specify the policy for a single Web document, portions of a Web site, or for an entire site. The policy reference file may refer to one or more P3P policies; this allows for a single reference file to cover an entire site, even if different P3P policies apply to different portions of the site.The policy reference file is used to make any or all of the following statements:

- The URI where a P3P policy is found
- The URIs or regions of URI-space covered by this policy
- The URIs or regions of URI-space not covered by this policy
- The regions of URI-space for embedded content on other servers that are covered by this policy
- The cookies that are or are not covered by this policy
- The access methods for which this policy is applicable
- The period of time for which these claims are considered to be valid

All of these statements are made in the body of the policy reference file.

2.2 Locating Policy Reference Files

This section describes the mechanisms used to indicate the location of a policy reference file. Detailed syntax is also given for the supported mechanisms.

The location of the policy reference file can be indicated using one of four mechanisms. The policy reference file

1. may be located in a predefined "well-known" location, or
2. a document may indicate a policy reference file through an HTML `link` tag, or
3. a document may indicate a policy reference file through an XHTML `link` tag, or
4. through an HTTP header.

Note that if user agents support retrieving HTML (resp. XHTML) content over HTTP, they MUST handle mechanisms 1, 2 and 3 (resp. 4) listed above interchangeably. See also the requirements for non-ambiguity.

Policies are applied at the level of resources. A "page" from the user's perspective may be composed of multiple HTTP resources; each may have its own P3P policy associated with it. As a practical note, however, placing many different P3P policies on different resources on a single page may make rendering the page and informing the user of the relevant policies difficult for user agents. Additionally, services are recommended to attempt to craft their policy reference files such that a single policy reference file covers any given "page"; this will speed up the user's browsing experience.

For a user agent to process the policy that applies to a given resource, it must locate the policy reference file for that resource, fetch the policy reference file, parse the policy reference file, fetch any required P3P policies, and then parse the P3P policy or policies.

This document does not specify how P3P policies may be associated with Web resources retrieved by means other than HTTP. However, it does not preclude future development of mechanisms for associating P3P policies with resources retrieved using other protocols. Furthermore, additional methods of associating P3P policies with HTTP resources may be developed in the future.

2.2.1 Well-Known Location

Web sites using P3P MAY (and, are strongly encouraged to) place a policy reference file in a "well-known" location. To do this, a policy reference file would be made available on the site at the path `/w3c/p3p.xml`.

Note that sites are not required to use this mechanism; however, by using this mechanism, sites can ensure that their P3P policy will be accessible to user agents before any other resources are requested from the site. This will reduce the need for user agents to access the site using safe zone practices. Additionally, if a site chooses to use this mechanism, the policy reference file located in the well-known location is not required to cover the entire site. For example, sites where not all of the content is under the control of a single organization MAY choose not to use this mechanism, or MAY choose to post a policy reference file which covers only a limited portion of the site.

Use of the well-known location for a policy reference file does not preclude use of other mechanisms for specifying a policy reference file. Portions of the site MAY use any of the other supported mechanisms to specify a policy reference file, so long as the non-ambiguity requirements are met.

For example, imagine a shopping-mall Web site run by the MallExample company. On their Web site (`mall.example.com`), companies offering goods or services at the mall would get a company-specific subtree of the site, perhaps in the path `/companies/`*`company-name`*. The MallExample company may choose to put a policy reference file in the well-known location which covers all of their site except the `/companies` subtree. Then if the ShoeStoreExample company has some content in `/companies/shoestoreexample`, they could use one of the other mechanisms to indicate the location of a policy reference file covering their portion of the `mall.example.com` site.

One case where using the well-known location for policy reference files is expected to be particularly useful is in the case of a site which has divided its content across several hosts. For example, consider a site which uses a different logical host for all of its Web-based applications than for its static HTML content. The other mechanisms allowed for specifying the location of a policy reference file require that some URI on the host being accessed must be fetched to locate the policy reference file. However, the well-known location mechanism has no such requirement. Consider the example of an HTML form located on `www.example.com`. Imagine that the action URI on that form points to server `cgi.example.com`. The policy reference file that covers the form is unable to make any statements about the action URI that processes the form. However, the site administrator publishes a policy reference file at `http://cgi.example.com/w3c/p3p.xml` that covers the action URI, thus enabling a user agent to easily locate the P3P policy that applies to the action URI before submitting the form contents.

2.2.2 HTTP Headers

Any document retrieved by HTTP MAY point to a policy reference file through the use of a new response header, the `P3P` header ([P3P-HEADER]). If a site is using P3P headers, it SHOULD include this on responses for all appropriate request methods, including `HEAD` and `OPTIONS` requests.

The P3P header gives one or more comma-separated directives. The syntax follows:

```
[1]   p3p-header             =    `P3P: ` p3p-header-field *(`,` p3p-header-field)
[2]   p3p-header-field       =    policy-ref-field | compact-policy-field | extension-field
[3]   policy-ref-field       =    `policyref="` URI-reference `"`
[4]   extension-field        =    token
                                  [`=` (token | quoted-string) ]
```

Here, `URI-reference` is defined as per RFC 2396 [URI], `token` and `quoted-string` are defined by [HTTP1.1].

In keeping with the rules for other HTTP headers, the name of the P3P header may be written with any casing. The contents should be specified using the casing precisely as specified in this document.

The `policyref` directive gives a URI which specifies the location of a policy reference file which may reference the P3P policy covering the document that pointed to the reference file, and possibly others as well. When the `policyref` attribute is a relative URI, that URI is interpreted relative to the request URI. Note that fetching the URI given in the `policyref` directive MAY result in a 300-class HTTP return code (redirection); user agents MUST interpret those redirects with normal HTTP semantics. Services should note, of course, that use of redirects will increase the time required for user agents to find and interpret their policies. The `policyref` URI MUST NOT be used for any other purpose beyond locating and referencing P3P policies.

The `compact-policy-field` is used to specify "compact policies". This is described in Section 4.

User agents which find unrecognized directives (in the `extension-field`s) MUST ignore the unrecognized directives. This is to allow easier deployment of future versions of P3P.

Example 2.1:

1. Client makes a `GET` request.

```
GET /index.html HTTP/1.1
Host: catalog.example.com
Accept: */*
Accept-Language: de, en
User-Agent: WonderBrowser/5.2 (RT-11)
```

2. Server returns content and the `P3P` header pointing to the policy of the resource.

```
HTTP/1.1 200 OK
P3P: policyref="http://catalog.example.com/P3P/PolicyReferences.xml"
Content-Type: text/html
Content-Length: 7413
Server: CC-Galaxy/1.3.18
```

2.2.3 The HTML `link` Tag

Servers MAY serve HTML content with embedded `link` tags (cf. [HTML]) that indicate the location of the relevant P3P policy reference file. This use of P3P does not require any change in the server behavior.

The `link` tag encodes the policy reference information that could be expressed using the `P3P` header. The `link` tag takes the following form (here, we just produce one possible ABNF format for the link tag, and suppose the [HTML] syntax rules can be used when using such a tag into an HTML file):

```
[5]   p3p-link-tag     =     `<link rel="P3Pv1" href="` URI `">`
```

Here, `URI` is defined as per RFC 2396 [URI].

When the `href` attribute is a relative URI, that URI is interpreted relative to the request URI.

In order to illustrate with an example the use of the `link` tag, we consider the policy reference expressed in Example 2.1 using HTTP headers. That example can be equivalently expressed using the link tag with the following piece of HTML:

```
<link rel="P3Pv1"
    href="http://catalog.example.com/P3P/PolicyReferences.xml">
```

Finally, note that since the `p3p-link-tag` is embedded in an HTML document, its character encoding will be the same as that of the HTML document. In contrast to P3P policy and policy reference documents (see section 2.3 and section 3 below), the `p3p-link-tag` need not be encoded using [UTF-8]. Note also that the `link` tag is not case sensitive.

2.2.4 The XHTML `link` tag

Correspondingly to the HTML `link` tag, P3P also supports XHTML (cf. [XHTML-MOD]). Servers MAY serve XHTML content that, using the *XHTML Link Module* (cf. Section 5.19 of [XHTML-MOD]), indicates the location of the relevant P3P policy reference file with an embedded XHTML `link` tag. Like in the HTML case, an XHTML `link` tag can be used to encode the policy reference information that could be expressed using the `P3P` header, by:

- setting its `rel` attribute to "`P3Pv1`"
- setting its `href` attribute to the URI of the relevant P3P policy reference file

2.2.5 HTTP ports and other protocols

The mechanisms described here MAY be used for HTTP transactions over any underlying protocol. This includes plain-text HTTP over TCP/IP connections or encrypted HTTP over SSL connections, as well as HTTP over any other communications protocol designers wish to implement.

URIs MAY contain network port numbers, as specified in RFC 2396 [URI]. For the purposes of P3P, different ports on a single host MUST be considered to be separate "sites". Thus, for example, the policy reference file at the well-known location for www.example.com on port 80 (http://www.example.com/w3c/p3p.xml) would not give any information about the policies which apply to www.example.com when accessed over SSL (as the SSL communication would take place on a different port, 443 by default).

This document does not specify how P3P policies may be associated with resources retrieved by means other than HTTP. However, it does not preclude future development of mechanisms for associating P3P policies with resources retrieved over other protocols. Furthermore, additional methods of associating P3P policies with resources retrieved using HTTP may be developed in the future.

2.3 Policy Reference File Syntax and Semantics

This section explains the contents of policy reference files in detail.

2.3.1 Example Policy Reference File

Consider the case of a Web site wishing to make the following statements:

1. P3P policy `/P3P/Policies.xml#first` applies to the entire site, except resources whose paths begin with `/catalog`, `/cgi-bin`, or `/servlet`.
2. P3P policy `/P3P/Policies.xml#second` applies to all resources whose paths begin with `/catalog`.
3. P3P policy `/P3P/Policies.xml#third` applies to all resources whose paths begin with `/cgi-bin` or `/servlet`, except for `/servlet/unknown`.
4. No statement is made about what P3P policy applies to `/servlet/unknown`.
5. These statements are valid for 2 days.

These statements can be represented by the following XML:

Example 2.2:

```
<META xmlns="http://www.w3.org/2002/01/P3Pv1">
 <POLICY-REFERENCES>
  <EXPIRY max-age="172800"/>

    <POLICY-REF about="/P3P/Policies.xml#first">
      <INCLUDE>/*</INCLUDE>
      <EXCLUDE>/catalog/*</EXCLUDE>
      <EXCLUDE>/cgi-bin/*</EXCLUDE>
      <EXCLUDE>/servlet/*</EXCLUDE>
    </POLICY-REF>

    <POLICY-REF about="/P3P/Policies.xml#second">
      <INCLUDE>/catalog/*</INCLUDE>
    </POLICY-REF>

    <POLICY-REF about="/P3P/Policies.xml#third">
      <INCLUDE>/cgi-bin/*</INCLUDE>
      <INCLUDE>/servlet/*</INCLUDE>
      <EXCLUDE>/servlet/unknown</EXCLUDE>
    </POLICY-REF>

 </POLICY-REFERENCES>
</META>
```

Note this example also includes via `EXPIRY` a relative expiry time in the document (cf. Section 2.3.2.3.2).

2.3.2 Policy Reference File Definition

This section defines the syntax and semantics of P3P policy reference files. All policy reference files MUST be encoded using [UTF-8]. P3P servers MUST encode their policy reference files using this syntax.

2.3.2.1 Policy reference file processing

2.3.2.1.1 Significance of order

A policy reference file has the META element as root. It may contain multiple POLICY-REF elements. If it does contain more than one element, they MUST be processed by user agents in the order given in the file. When a user agent is attempting to determine what policy applies to a given URI, it MUST use the first POLICY-REF element in the policy reference file which applies to that URI.

Note that each POLICY-REF may contain multiple INCLUDE, EXCLUDE, METHOD, COOKIE-INCLUDE, and COOKIE-EXCLUDE elements and that all of these elements within a given POLICY-REF MUST be considered together to determine whether the POLICY-REF applies to a given URI. Thus, it is not sufficient to find an INCLUDE element that matches a given URI, as EXCLUDE or METHOD elements may serve as modifiers that cause the POLICY-REF not to match.

2.3.2.1.2 Wildcards in policy reference files

Policy reference files make statements about what policy applies to a given URI. Policy reference files support a simple wildcard character to allow making statements about regions of URI-space. The character asterisk ('*') is used to represent a sequence of 0 or more of any character. No other special characters (such as those found in regular expressions) are supported.

Note that since the asterisk is also a legal character in URIs ([URI]), some special conventions have to be followed when encoding such "extended URIs" in a policy reference file:

- URIs represented in policy reference files MUST be properly escaped, as described in [URI], *except*:
  - Literal '*'s in URIs MUST be escaped in policy reference files (i.e., they MUST be represented as "%2A"). Any '*' present in a URI within a policy reference file will be taken as representing the asterisk wildcard character.
  - Consequently, P3P user agents MUST properly un-escape a URI given in a policy reference file, according to [URI], before trying to match it against an internally represented URI, but only after recognizing any literal '*' present as the asterisk wildcard character.

URI escaping and unescaping is very much dependant on the actual scheme used, and might even differ between individual components within a single scheme, so no simple rule for which characters need to be escaped can be given here. Please refer directly to [URI] for details on the standard escaping process. Note that P3P user agents MAY ignore any URI pattern that does not conform to [URI].

The wildcard character MAY be used in the INCLUDE and EXCLUDE elements, in the COOKIE-INCLUDE and COOKIE-EXCLUDE elements, and in the HINT element.

2.3.2.2 The META and POLICY-REFERENCES elements

<META>

The META element contains a complete policy reference file. Optionally, one POLICIES element can follow. META can also contain one or more one or more EXTENSION elements (cf. section 3.5), as well as an xml:lang attribute (see section 2.4.2), to indicate the language in which its content is expressed.

<POLICY-REFERENCES>

This element MAY contain one or more POLICY-REF (policy reference) elements. It MAY also contain one EXPIRY element (indicating their expiration time), one or more HINT element, and one or more EXTENSION element (cf. section 3.5).

```
[6]    prf           =    `<META xmlns="http://www.w3.org/2002/01/P3Pv1"` [xml-lang] `>`
                          *extension
                          policyrefs
                          [policies]
                          *extension
                          "</META>"

[7]    policyrefs    =    "<POLICY-REFERENCES>"
                          [expiry]
                          *policyref
                          *hint
                          *extension
                          "</POLICY-REFERENCES>"
```

Here PCDATA is defined in [XML].

2.3.2.3 Policy reference file lifetimes and the EXPIRY element

2.3.2.3.1 Motivation and mechanism

It is desirable for servers to inform user agents about how long they can use the claims made in a policy reference file. By enabling clients to cache the contents of a policy reference file, it reduces the time required to process the privacy policy associated with a Web resource. This also reduces load on the network. In addition, clients that don't have a valid policy reference file for a URI will need to use "safe zone" practices for their requests. If clients have policy reference files that they know are still valid, then they can make more informed decisions on how to proceed.

In order to achieve these benefits, policy reference files SHOULD contain an EXPIRY element, which indicates the lifetime of the policy reference file. If the policy reference file does not contain an EXPIRY element, then it defaults to 24-hour lifetime.

The lifetime of a policy reference file tells user agents how long they can rely on the claims made in the policy reference file. By setting the lifetime of a policy reference file, the publishing site agrees that the policies mentioned in the policy reference file are appropriate for the lifetime of the policy reference file. For example, if a policy reference file has a lifetime of 3 days, then a user agent need not reload that file for 3 days, and can assume that the references made in that policy reference file are good for 3 days. All of the policy references made in a single policy reference file will receive the same lifetime. The only way to specify different lifetimes for different policy references is to use separate policy reference files.

The same mechanism used to indicate the lifetime of a policy reference file is also used to indicate the lifetime of a P3P policy. Thus P3P POLICIES elements SHOULD have an EXPIRY element associated with them as well. This lifetime applies to all P3P policies contained within that POLICIES element. If there is no EXPIRY element associated with a P3P policy, then it defaults to 24-hour lifetime.

When picking a lifetime for policies and policy reference files, sites need to pick a lifetime which balances two competing concerns. One concern is that the lifetime ought to be long enough to allow user agents to receive significant benefits from caching. The other concern is that the site would like to be able to change their policy for new data collection without waiting for an extremely long lifetime to expire. It is expected that lifetimes in the range of 1-7 days would be a reasonable balance between these two competing desires. Sites also need to remember the policy update requirements when updating their policies.

When a policy reference file has expired, the information in the policy reference file MUST NOT be used by a user agent until that user agent has successfully revalidated the policy reference file, or has fetched a new copy of the policy reference file.

Note that while user agents are not obligated to revalidate policy reference files or policy files that have not expired, they MAY choose to revalidate those files before their expiry period has passed in order to reduce the need for using "safe zone" practices. A valid P3P user agent implementation does not need to contain a cache for policies and policy reference files, though the implementation will have better performance if it does.

2.3.2.3.2 The `EXPIRY` element

The `EXPIRY` element can be used in a policy reference file and/or in a `POLICIES` element to state how long the policy reference file (or policies) remains valid. The expiry is given as either an absolute expiry time, or a relative expiry time. An absolute expiry time is a time, given in GMT, until which the policy reference file (or policies) is valid. A relative expiry time gives a number of seconds for which the policy reference file (or policies) is valid. This expiry time is relative to the time the policy reference file (or policies) was requested or last revalidated by the client. This computation MUST be done using the time of the original request or revalidation, and the current time, with both times generated from the client's clock. Revalidation is defined in section 13.3 of [HTTP1.1].

The minimum amount of time for any relative expiry time is 24 hours, or 86400 seconds. Any relative expiration time shorter than 86400 seconds MUST be treated as being equal to 86400 seconds in a client implementation. If a client encounters an absolute expiration time that is in the past, it MUST act as if NO policy reference file (or policy) is available. See section 2.4.7 "Absence of Policy Reference File" for the required procedure in such cases.

```
[8]   expiry      =   "<EXPIRY" (absdate|reldate) "/>"
[9]   absdate     =   `date="` HTTP-date `"`
[10]  reldate     =   `max-age="` delta-seconds `"`
```

Here, HTTP-date is defined in section 3.3.1 of [HTTP1.1], and delta-seconds is defined in section 3.3.2 of [HTTP1.1].

2.3.2.3.3 Requesting Policies and Policy Reference Files

In a real-world network, there may be caches which will cache the contents of policies and policy reference files. This is good for increasing the overall network performance, but may have deleterious effects on the operation of P3P if not used correctly. There are two specific concerns:

1. When a user agent receives a policy reference file (or policy), if it was served from a caching proxy (see e.g. [CACHING]) the user agent needs to know how long the policy reference file or policy resided in the caching proxy. This time MUST be subtracted from the lifetime of the policy or policy reference file which uses relative expiry.
2. When a user agent needs to revalidate a policy reference file (or policy), it needs to make sure that the revalidation fetches a current version of the policy reference file (or policy). For example, consider the case where a user agent holds a policy reference file with a 1 day relative expiry. If the user agent refetches it from a caching proxy, and the file has been residing in the caching proxy for 3 days, then the resulting file is useless.

HTTP 1.1 [HTTP1.1] contains powerful cache-control mechanisms to allow clients to place requirements on the operations of network caches; these mechanisms can resolve the problems mentioned above. The specific method will be discussed below.

HTTP 1.0, however, does not provide those more sophisticated cache control mechanisms. An HTTP 1.0 caching proxy will, in all likelihood, compute a cache lifetime for the policy reference file (or policies) based on the file's last-modified date; the resulting cache lifetime could be significantly longer than the lifetime specified by the `EXPIRY` element. The caching proxy could then serve the policy reference file (or policies) to clients beyond the lifetime in the `EXPIRY`; the result would be that user-agents would receive a useless policy reference file (or policies).

The second problem with an HTTP 1.0 caching proxy is that a user agent has no way to know how long the reference file may have been stored by the caching proxy. If the policy reference file (or policies) relies on relative expiry, it would then be impossible for the user agent to determine if the reference file's lifetime has already expired, or when it will expire.

Thus, if a user agent is requesting a policy reference file or a policy, and does not know for certain that there are no HTTP 1.0 caches in the path to the origin server, then the request MUST force an end-to-end revalidation. This can be done with the `Pragma: no-cache` HTTP request-header. Note that neither HTTP nor P3P define a way to determine if there is a HTTP 1.0-compliant cache in any given network path, so unless the user agent has this information derived from an outside source, it MUST force the end-to-end revalidation.

If the user agent has some way to know that all caches in the network path to the origin server are compliant with HTTP 1.1 (or that there are no caches in the network path to the origin server), then the client MAY do the following instead of forcing an end-to-end revalidation:

1. Use cache-control request-headers to ensure that the received response is not older than its lifetime. This is done with the max-age cache-control setting, with a maximum age significantly less than the lifetime of the policy reference file (or policies). For example, a user agent could send Cache-Control: max-age=43200, thus ensuring that the response is no more than 12 hours old.
2. Subtract the age of the response from the lifetime of the policy reference file (or policies), if it uses a relative expiry time. The age of the response is given by the Age: HTTP response-header.

Note that it is impossible for a client to accurately predict the amount of latency that may affect an HTTP request. Thus, if the policy reference file covering a request is going to expire soon, clients MAY wish to consider warning their users and/or revalidating the policy reference file before continuing with the request.

2.3.2.3.4 Error handling for policy reference file and policy lifetimes

The following situations have their semantics specifically defined:

1. An absolute expiry date in the past renders the policy reference file (or policies) useless, as does an invalid or malformed expiry date, whether relative or absolute. In this case, user agents MUST act as if NO policy reference file (or policies) is available. See section 2.4.7 "Absence of Policy Reference File" for the required procedure in such cases.
2. A relative expiration time shorter than 86400 seconds (1 day) is considered to be equal to 86400 seconds.
3. When a policy reference file contains more than one `EXPIRY` element, the first one takes precedence for determining the lifetime of the policy reference file.

2.3.2.4 The `POLICY-REF` element

A policy reference file may refer to multiple P3P policies, specifying information about each. The `POLICY-REF` element describes attributes of a single P3P policy. Elements within the `POLICY-REF` element give the location of the policy and specify the areas of URI-space (and cookies) that each policy covers.

`POLICY-REF`
: contains information about a single P3P policy.

`about` ***(mandatory attribute)***
: *URI reference* ([URI]), where the fragment identifier part denotes the *name* of the policy (given in its `name` attribute), and the URI part denotes the URI where the policy resides (a policy file, or a policy reference file, see Section 3.2). If this is a relative URI reference, it is interpreted relative to the URI of the policy reference file it resides in.

```
[11]    policy-ref              =        `<POLICY-REF about="` URI-reference `">`
                                         *include
                                         *exclude
                                         *cookie-include
                                         *cookie-exclude
                                         *method-element
                                         *extension
                                         `</POLICY-REF>`
```

Here, `URI-reference` is defined as per RFC 2396 [URI].

2.3.2.5 The `INCLUDE` and `EXCLUDE` elements

Each `INCLUDE` or `EXCLUDE` element specifies one local URI or set of local URIs. A set of URIs is specified if the wildcard character '*' is used in the URI-pattern. These elements are used to specify the portion of the Web site that is covered by the policy referenced by the enclosing `POLICY-REF` element.

When `INCLUDE` (and optionally, `EXCLUDE`) elements are present in a `POLICY-REF` element, it means that the policy specified in the `about` attribute of the `POLICY-REF` element applies to all the URIs at the requested host corresponding to the local-URI(s) matched by any of the `INCLUDE`s, but not matched by an `EXCLUDE` element.

A policy referenced in a policy reference file can be applied only to URIs on the DNS (Domain Name System) host that references it. Thus, for example, a policy reference file at the well-known location of host www.example.com can apply policies only to resources on www.example.com. However, if foo.example.com includes a P3P HTTP header in its responses that references a policy reference file on bar.example.com, that policy reference file would be applied to resources on foo.example.com (not bar.example.com or www.example.com). The same policy reference file might be referenced in P3P HTTP headers sent by multiple hosts, in which case it may be applied to each host that references it. The `INCLUDE` and `EXCLUDE` elements MUST specify URI patterns relative to the root of the DNS host to which they are applied. This requirement does NOT apply to the location of the P3P policy file (the about attribute on the `POLICY-REF` element).

If a `METHOD` element (section 2.3.2.8) specifies one or more methods for an enclosing policy reference, it follows that all methods *not* mentioned are consequently *not* covered by this policy. In the case that this is the only policy reference for a given URI prefix, user agents MUST assume that NO policy is in effect for all methods NOT mentioned in the policy reference file. It is legal but pointless to supply a `METHOD` element without any `INCLUDE` or `COOKIE-INCLUDE` elements.

It is legal, but pointless, to supply an `EXCLUDE` element without any `INCLUDE` elements; in that case, the `EXCLUDE` element MUST be ignored by user agents.

Note that the set of URIs specified with `INCLUDE` and `EXCLUDE` does not include cookies that might be set or replayed when requesting one of such URIs: in order to associate policies with cookies, the `COOKIE-INCLUDE` and `COOKIE-EXCLUDE` elements are needed.

```
[12]    include              =          "<INCLUDE>" relativeURI "</INCLUDE>"
[13]    exclude              =          "<EXCLUDE>" relativeURI "</EXCLUDE>"
```

Here, `relativeURI` is defined as per RFC 2396 [URI], with the addition that the '`*`' character is to be treated as a wildcard, as defined in section 2.3.2.1.2.

2.3.2.6 The `HINT` element

Policy reference hints are a performance optimization that can be used under certain conditions. A site may declare a policy reference for itself using the well-known location, the P3P response header, or the HTML/XHTML `link` tag. It MAY further provide a hint to additional policy references, such as those declared by other sites.

For example, an HTML page might hint at policy references for its hyperlinks, embedded content, and action URIs. User agents MAY use the hint mechanism to discover policy reference files before requesting the affected URIs when the policy references are not available from the well-known location.

User agents which use hints to retrieve policies MUST NOT apply them to any site other than the one which contains the hinted policy reference file.

Any policy reference file MAY contain zero or more policy reference hints. Each hint is contained in a `HINT` element with two attributes, `scope` and `path`.

The `scope` attribute is used to specify a URI scheme and authority to which the hinted policy reference can be applied. If the authority component (cf. [URI]) is a server component (e.g., a hostname or IP address) the host part of the authority MAY begin with a wildcard, as defined in Section 2.3.2.1.2. The `scope` attribute MUST NOT contain a wildcard in any other position, MUST be encoded according to the conventions in Section 2.3.2.1.2, and MUST NOT contain a path, query or fragment URI component. Additionally, if the authority is a server, it SHOULD NOT contain a userinfo part.

For example, legal values for `scope` include:

- `http://www.example.com`
- `http://www.example.com:81`
- `http://*.example.com`
- `ftp://ftp.example.org`

The following are illegal values for the `scope` attribute:

- `http://www.*.com` ; the wildcard can only be at the start
- `http://www.example.com/` ; the trailing slash is not allowed
- `www.example.com` ; the scheme must be stated
- `*://www.example.com` ; the scheme cannot contain a wildcard
- `http://www.example.com:*`; the port cannot contain a wildcard

The `path` attribute is used to locate the policy reference file on the hinted site. It is a relative URI whose base is the URI scheme and authority matched in the `scope` attribute. The `path` attribute MUST NOT be an absolute URI, so that the policy reference file is always retrieved from the same site that it is applied to.

Example 2.3:

```
<HINT scope="http://www.example.org" path="/mypolicy/p3.xml" />
<HINT scope="http://www.example.net:81" path="/w3c/prf.xml" />
<HINT scope="http://*.shop.example.com" path="/w3c/prf.xml" />
```

```
[14]   hint      =     `<HINT scope="` scheme ( `://` | `:/` ) authority `" path="` relativeURI `/>`
```

Here, `scheme`, `authority` and `relativeURI` are taken from RFC 2965 [STATE].

2.3.2.7 The `COOKIE-INCLUDE` and `COOKIE-EXCLUDE` elements

The `COOKIE-INCLUDE` and `COOKIE-EXCLUDE` elements are used to associate policies to cookies (cf. [COOKIES] and [STATE]).

A cookie policy MUST cover any data (within the scope of P3P) that is stored in that cookie or linked via that cookie. It MUST also reference all purposes associated with data stored in that cookie or enabled by that cookie. In addition, any data/purpose stored or linked via a cookie MUST also be put in the cookie policy. In addition, if that linked data is collected by HTTP, then the policy that covers that `GET`/`POST`/whatever request must cover that data collection. For example, when CatalogExample asks customers to fill out a form with their name, billing, and shipping information, the P3P policy that covers the form submittal will disclose that CatalogExample collects this data and explain how it is used. If CatalogExample sets a cookie so that it can recognize its customers and observe their behavior on its Web site, it would have a separate policy for this cookie. However, if this cookie is also linked to the user's name, billing, and shipping information -- perhaps so CatalogExample can generate custom catalog pages based on where the customer lives -- then that data must also be disclosed in the cookie policy.

For the purpose of this specification, state management mechanisms use either `SET-COOKIE` or `SET-COOKIE2` headers, and cookie-namespace is defined as the value of the NAME, VALUE, Domain and Path attributes, specified in [COOKIES] and [STATE].

Each `COOKIE-INCLUDE` or `COOKIE-EXCLUDE` element can be used to match (similarly to `INCLUDE` and `EXCLUDE`) the NAME, VALUE, Domain and Path components of a cookie, expressing the cookies which are covered by the policy specified by the `about` attribute when the cookies are set from the resources on the Web site where the policy reference file resides:

`COOKIE-INCLUDE` (resp. `COOKIE-EXCLUDE`)
: include (resp. exclude) cookies that match the `name`, `value`, `domain` and `path` attributes

`name`
: match the NAME portion of the cookie

`value`
: match the VALUE portion of the cookie

`domain`
: match the Domain portion of the cookie

`path`
: match the Path portion of the cookie

If the value of the `domain` attribute is set to the dot character ("`.`"), the domain will match only cookies that omit the `domain` attribute (and thus have domain equivalent to the request host as per RFC 2965 ([STATE]).

Cookies that omit the path attribute have the default path of the request URI that generated the set-cookie response as per RFC 2965 [STATE]. The `path` attribute of a `COOKIE-INCLUDE` should be matched against this default value if a cookie omits the `path` attribute.

All four attributes are optional. If an attribute is absent, the `COOKIE-INCLUDE` (resp. `COOKIE-EXCLUDE`) will match cookies that have that attribute set to any value.

When `COOKIE-INCLUDE` (and optionally, `COOKIE-EXCLUDE`) elements are present in a `POLICY-REF` element, the policy specified in the `about` attribute of the `POLICY-REF` element applies to every cookie that is matched by any `COOKIE-INCLUDE`'s, and not matched by a `COOKIE-EXCLUDE` element.

User agents MUST interpret `COOKIE-INCLUDE` and `COOKIE-EXCLUDE` elements in a policy reference file to determine the policy that applies to cookies set by or replayed to the host to which the policy reference file applies. While the domain attribute of a `COOKIE-INCLUDE` may match more broadly (for example, if the domain attribute is omitted it defaults to matching any domain value), user agents MUST limit their application of the policy to domains that could be legally used in a cookie set by the host to which the policy reference file applies. For example, if abc.xyz.example.com declares a policyref with `<COOKIE-INCLUDE domain="*.xyz.*ple.com"/>`, this would be matched to cookies with domains such as .abc.xyz.example.com and .xyz.example.com, but not .example.com or .xyz.sample.com.

A P3P policy can be associated with a cookie by the host that set that cookie as well as by any or all of the hosts to which it might be replayed. A user agent MAY fetch a cookie policy at the time a cookie is set and apply it later when the cookie is replayed, perhaps to other hosts in the domain. A user agent MAY request a policy reference file from a host before replaying a cookie to that host, and if the policy reference file contains an appropriate `COOKIE-INCLUDE`, a policy will be applied to that cookie even if the cookie was not set by that host. Any host to which the cookie may be replayed MUST be able to honor all the policies associated with the cookie, regardless of whether that host declares a policy for that cookie. Thus sites that set cookies that may be replayed to multiple hosts within a domain need to coordinate to make sure all the hosts can follow the declared policy. In addition, sites should be cautious with their use of wildcards to make sure that they do not inadvertently apply a policy to cookies to which it should not be applied (including previously set cookies that are still in use and cookies set by other hosts in the domain).

The policy that applies to a cookie applies until the policy expires, even if the associated policy reference file expires prior to policy expiry (but after the cookie was set). If the policy associated with a cookie has expired, then the user agent SHOULD reevaluate the cookie policy before sending the cookie. In addition, user agents MUST use only non-expired policies and policy reference files when evaluating new set-cookie events.

Example 2.4 states that `/P3P/Policies.xml#first` applies to all cookies.

Example 2.4:

```
<META xmlns="http://www.w3.org/2002/01/P3Pv1">
 <POLICY-REFERENCES>
    <POLICY-REF about="/P3P/Policies.xml#first">
       <COOKIE-INCLUDE name="*" value="*" domain="*" path="*"/>
    </POLICY-REF>
 </POLICY-REFERENCES>
</META>
```

Example 2.5 states that `/P3P/Policies.xml#first` applies to all cookies, except cookies with the cookie name value of "`obnoxious-cookie`", a domain value of "`.example.com`", and a path value of "`/`", and that `/P3P/Policies.xml#second` applies to all cookies with the cookie name of "`obnoxious-cookie`", a domain value of "`.example.com`", and a path value of "`/`".

Example 2.5:

```
<META xmlns="http://www.w3.org/2002/01/P3Pv1">
 <POLICY-REFERENCES>
    <POLICY-REF about="/P3P/Policies.xml#first">
       <COOKIE-INCLUDE name="*" value="*" domain="*" path="*"/>
       <COOKIE-EXCLUDE name="obnoxious-cookie" value="*" domain=".example.com" path="/"/>
    </POLICY-REF>
    <POLICY-REF about="/P3P/Policies.xml#second">
       <COOKIE-INCLUDE name="obnoxious-cookie" value="*" domain=".example.com" path="/"/>
    </POLICY-REF>
 </POLICY-REFERENCES>
</META>
```

```
[15]  cookie-include     =    "<COOKIE-INCLUDE"
                                 [` name="` token `"`]   ; matches the cookie's NAME
                                 [` value="` token `"`]  ; matches the cookie's VALUE
                                 [` domain="` token `"`] ; matches the cookie's Domain
                                 [` path="` token `"`]   ; matches the cookie's Path
                              "/>"

[16]  cookie-exclude     =    "<COOKIE-EXCLUDE"
                                 [` name="` token `"`]   ; matches the cookie's NAME
                                 [` value="` token `"`]  ; matches the cookie's VALUE
                                 [` domain="` token `"`] ; matches the cookie's Domain
                                 [` path="` token `"`]   ; matches the cookie's Path
                              "/>"
```

Here, `token`, `NAME`, `VALUE`, `Domain` and `Path` are defined as per RFC 2965 [STATE], with the addition that the '`*`' character is to be treated as a wildcard, as defined in section 2.3.2.1.2.

Note that [STATE] states default values for the domain and path attributes of cookies: these should be used in the comparison if those attributes are not found in a specific cookie. Also, conforming to [STATE], if an explicitly specified `Domain` value does not start with a full stop ("`.`"), the user agent MUST prepend a full stop for it; and, note that every `Path` begins with the "`/`" character.

2.3.2.8 The `METHOD` element

By default, a policy reference applies to the stated URIs regardless of the method used to access the resource. However, a Web site may wish to define different P3P policies depending on the method to be applied to a resource. For example, a site may wish to collect more data from users when they are performing `PUT` or `DELETE` methods than when performing `GET` methods.

The `METHOD` element in a policy reference file is used to state that the enclosing policy reference only applies when the specified methods are used to access the referenced resources. The `METHOD` element may be repeated to indicate multiple applicable methods. If the `METHOD` element is not present in a `POLICY-REF` element, then that `POLICY-REF` element covers the resources indicated regardless of the method used to access them.

So, to state that `/P3P/Policies.xml#first` applies to all resources whose paths begin with `/docs/` for `GET` and `HEAD` methods, while `/P3P/Policies.xml#second` applies for `PUT` and `DELETE` methods, the following policy reference would be written:

Example 2.6:

```
<META xmlns="http://www.w3.org/2002/01/P3Pv1">
 <POLICY-REFERENCES>
    <POLICY-REF about="/P3P/Policies.xml#first">
      <INCLUDE>/docs/*</INCLUDE>
      <METHOD>GET</METHOD>
      <METHOD>HEAD</METHOD>
    </POLICY-REF>
    <POLICY-REF about="/P3P/Policies.xml#second">
      <INCLUDE>/docs/*</INCLUDE>
      <METHOD>PUT</METHOD>
      <METHOD>DELETE</METHOD>
    </POLICY-REF>
 </POLICY-REFERENCES>
</META>
```

Note that HTTP requires the same behavior for `GET` and `HEAD` requests, thus it is inappropriate to specify different P3P policies for these methods. The syntax for the `METHOD` element is:

[17]

```
method-element          =          `<METHOD>` Method `</METHOD>`
```

Here, `Method` is defined in the section 5.1.1 of [HTTP1.1].

Finally, note that the `METHOD` element is designed to be used in conjunction with `INCLUDE` or `COOKIE-INCLUDE` elements. A `METHOD` element by itself will never apply a `POLICY-REF` to a URI.

2.3.3 Applying a Policy to a URI

A policy reference file specifies the policy which applies to a given URI. In other words, the indicated policy describes all effects of dereferencing the given URI (in some cases, with the appropriately specified `METHOD`).

There is a general rule which describes what it means for a P3P policy to cover a URI: *the referenced policy MUST cover actions that the user's client software is expected to perform as a result of requesting that URI*. Obviously, the policy must describe all data collection performed by site as a result of processing the request for the URI. Thus, if a given URI is covered for terms of `GET` requests, then the policy given by the policy reference file MUST describe all data collection performed by the site when that URI is dereferenced. Likewise, if a URI is covered for `POST` requests, then any data collection that occurs as a result of POSTing a form or other content to that URI MUST be described by the policy.

The concept of "actions that the client software is expected to perform" includes the setting of client-side cookies or other state-management mechanisms invoked by the response. If executable code is returned when a URI is requested, then the P3P policy covering that URI MUST cover certain actions which will occur when that code is executed. The covered actions are any actions which could take place without the user explicitly invoking them. If explicit user action causes data to be collected, then the P3P policy covering the URI for that action would disclose that data collection.

Some specific examples:

1. Fetching a URI returns an HTML page which contains a form, and the form contents are sent to a second URI when the user clicks a "Submit" button. The P3P policy covering the second URI MUST disclose all data collected by the form. The P3P policy covering the first URI (the URI the form was loaded from) MAY or MAY NOT disclose any of the data that will be collected on the form.
2. An HTML page includes JavaScript code which tracks how long the page is displayed and whether the user moved the mouse over a certain object on the page; when the page is unloaded, the JavaScript code sends that information to the server where the HTML page originated. The activity of the JavaScript code MUST be covered by the P3P policy of the HTML page. The reasoning is that this activity takes place without the user's knowledge or consent, and it occurs automatically as a result of loading the page.
3. A resource returns an executable for an electronic mail program. In order to use the email program, the user must run an installation program, start the email program, and use its facilities. The P3P policy covering URI from where the email program was downloaded is not required to make a statement about the data which could be collected by using the email program. Installing and running the email program is clearly outside the Web browsing experience, so it is not covered by this specification. A separate protocol could be designed to allow downloaded applications to present a P3P policy, but this is outside the scope of this specification.
4. An HTML page containing a form includes a reference to an executable which provides a custom client-side control. The data in the control is submitted to a site when the form is submitted. In this case, the URI for the HTML page and the URI for the custom control is not required to make a statement about the data the custom control represents. However, the URI to which the form contents are posted MUST cover the data from the custom control, just as it would cover any other data collected by processing the form. This behavior is similar to the way HTML forms are handled when they use only standard HTML controls: the control itself collects no data, and the data is collected when the form is posted. Note that this example assumes that the form is only posted when the user actively presses a "submit" or similar button. If the form were posted automatically (for example, by some JavaScript code in the page), then this example would be similar to example #2, and the data collected by the form MUST be described in the P3P policy which covers the HTML form.
5. Requests to a URI are redirected to a third party. If the first party embeds previously collected personal data in the query string or other part of the redirect URI, the privacy policy for the first party's URI MUST describe the types of data transmitted and include the third party as a recipient.

2.3.4 Forms and Related Mechanisms

Forms deserve special consideration, as they often link to CGI scripts or other server-side applications in their action URIs (the *action URI* is the URI given in the action attribute of the HTML `<FORM>` element, as defined in section 17.3 of [HTML]). It is often the case that those action URIs are covered by a different policy than the form itself.

If a user agent is unable to find a matching include-rule for a given action URI in the policy reference file that was referenced from the page, it SHOULD assume that *no* policy is in effect. Under these circumstances, user agents SHOULD check the well-known location on the host of the action URI to attempt to find a policy reference file that covers the action URI. If this does not provide a P3P policy to cover the action URI, then a user agent MAY try to retrieve the policy reference file by using the `HINT` mechanism on the action URI, and/or by issuing a `HEAD` request to the action URI before actually submitting any data in order to find the policy in effect. Services SHOULD ensure that server-side applications can properly respond to such `HEAD` requests and return the corresponding policy reference link in the headers. In case the underlying application does not understand the `HEAD` request and *no* policy has been predeclared for the action URI in question, user agents MUST assume that *no* policy is in effect and SHOULD inform the user about this or take the corresponding actions according to the user's preferences.

Note that services might want to make use of the `<METHOD>` element in order to declare policies for server-side applications that only cover a subset of supported methods, e.g., `POST` or `GET`. Under such circumstances, it is acceptable that the application in question only supports the methods given in the policy reference file (e.g., `PUT` requests need not be supported). User agents SHOULD NOT attempt to issue a `HEAD` request to an action URI if the relevant methods specified in the form's `method` attribute have been properly predeclared in the page's policy reference file.

In some cases, *different* data is collected at the *same* action URI depending on some selection in the form. For example, a search service might offer to both search for people (by name and/or email) and (arbitrary) images. Using a set of radio buttons on the form, a single server-side application located at one and the same action URI handles both cases and collects the required information necessary for the search. If a service wants to predeclare the data collection practices of the server-side application it MAY declare *all* of the data collection practices in a *single* policy file (using a `<INCLUDE>` declaration matching the action URI). In this case, user agents MUST assume that all data elements are collected under every circumstance. This solution offers the convenience of a single policy but might not properly reflect the fact that only parts of the listed data elements are collected at a time. Services SHOULD make sure that a simple `HEAD` request to the action URI (i.e., without any arguments, especially without the value of the selected radio button) will return a policy that covers all cases.

Note that if a form is handled through use of the `GET` method, then the action URI reflects the choice of form elements selected by the user. In some cases, it will be possible to make use of the wildcard syntax allowed in policy reference files to specify different policies for different uses of the same form action-handler URI. Therefore, user agents MUST include the query-string portion of URIs when making comparisons with INCLUDE and EXCLUDE elements in policy reference files.

2.4 Additional Requirements

2.4.1 Non-ambiguity

User agents need to be able to determine unambiguously what policy applies to a given URI. Therefore, sites SHOULD avoid declaring more than one non-expired policy for a given URI. In some rare case sites MAY declare more than one non-expired policy for a given URI, for example, during a transition period when the site is changing its policy. In those cases, the site will probably not be able to determine reliably which policy any given user has seen, and thus it MUST honor all policies (this is also the case for compact policies, cf. Section 4.1 and Section 4.6). Sites MUST be cautious in their practices when they declare multiple policies for a given URI, and ensure that they can actually honor all policies simultaneously.

If a policy reference file at the well-known location declares a non-expired policy for a given URI, this policy applies, regardless of any conflicting policy reference files referenced through HTTP headers or HTML/XHTML link tags.

If an HTTP response header includes references to more than one policy reference file, P3P user agents MUST ignore all references after the first one.

If an HTML (resp. XHTML) file includes HTML (resp. XHTML) `link` tag references to more than one policy reference file, P3P user agents MUST ignore all references after the first one.

If a user agent discovers more than one non-expired P3P policy for a given URI (for example because a page has both a P3P header and a `link` tag that reference different policy reference files, or because P3P headers for two pages on the site reference different policy reference files that declare different policies for the same URI), the user agent MAY assume any (or all) of these policies apply as the site MUST honor all of them.

2.4.2 Multiple Languages

Multiple language versions (translations) of the same policy can be offered by the server using the HTTP "`Content-Language`" header to properly indicate that a particular language has been used for the policy. This is useful so that human-readable fields such as entity and consequence can be presented in multiple languages. The same mechanism can also be used to offer multiple language versions for data schemas. Servers SHOULD return a localized policy in response to an HTTP request with an HTTP "`Accept-Language`" header when a policy matching the given language preferences is available.

Whenever `Content-Language` is used to distinguish policies at the same URI that are offered in multiple languages, the policies MUST have the same meaning in each language. Two policies (or two data schemas) are taken to be identical if

- All formal (not natural language) protocol elements are semantically identical (e.g., attribute order does not matter, the presence or absence of a default value does not matter, but attribute values matter)
- All natural language protocol elements correspond one-to-one, and for each correspondence, one is a careful translation of the other.

Due to the use of the `Accept-Language` mechanism, implementers should take note that user agents may see different language versions of a policy or policy reference file despite sending the same `Accept-Language` request header if a new language version of a policy or data schema has been added.

Finally, language declarations can be also included directly within P3P XML files: the `POLICY`, `POLICIES`, `META`, and `DATASCHEMA` elements MAY take an `xml:lang` attribute to indicate the language of any human-readable fields they contain (`xml:lang` is normatively defined in section 2.12 of [XML]).

```
[18]   xml-lang              =          ` xml:lang="` language `"`
```

Here, `language` is a language identifier as defined in [LANG].

2.4.3 The "Safe Zone"

P3P defines a special set of "safe zone" practices, which SHOULD be used by all P3P-enabled user agents and services for the communications which take place as part of fetching a P3P policy or policy reference file. In particular, requests to the well-known location for policy reference files SHOULD be covered by these "safe zone" practices. Communications covered by the safe zone practices SHOULD have only minimal data collection, and any data that is collected is used only in non-identifiable ways.

To support this safe zone, P3P user agents SHOULD suppress the transmission of data unnecessary for the purpose of finding a site's policy until the policy has been fetched. Therefore safe-zone practices for user agents include the following requirements:

- User agents SHOULD NOT send the HTTP `Referer` header in the safe zone
- User agents SHOULD NOT accept cookies from safe-zone requests
- User agents MAY also wish to refrain from sending user agent information or cookies accepted in a previous session on safe zone requests
- User agent implementers need to be aware that there is a privacy trade-off with using the `Accept-Language` HTTP header in the safe zone. Sending the correct `Accept-Language` header will allow fetching a P3P policy in the user's preferred natural language (if available), but does expose a certain amount of information about the identity of the user. User agents MAY wish to allow users to decide when these headers should be sent.

Safe-zone practices for servers include the following requirements:

- Servers SHOULD NOT require the receipt of an HTTP `Referer` header, cookies, user agent information, or other information unnecessary for responding to requests in the safe zone
- If the user agent suppresses the `Accept-Language` HTTP header as part of safe-zone operation, the server is free to choose any of the available translations
- If the communications is taking place over a secure connection (such as SSL), then the server SHOULD NOT require an identity certificate from the user agent for safe zone requests
- In addition, servers SHOULD NOT use in an identifiable way any information collected while serving a safe zone request

Note that the safe zone requirements do not say that sites cannot keep identifiable information -- only that they SHOULD NOT use in an identifiable way any information collected while serving a policy file or policy reference file. Tracking down the source of a denial of service attack, for example, would be a legitimate reason to use this information.

2.4.4 Policy and Policy Reference File Processing by User Agents

P3P user agents MUST only render or act upon P3P policies and policy reference files that are well-formed XML.

P3P user agents SHOULD only render or act upon P3P policies and policy reference files that conform to the XML schema given in Appendix 4, and user agents SHOULD NOT rely upon any part of a policy or policy reference file that does not conform to this XML schema.

User agents MUST NOT locally modify a P3P policy or policy reference file in order to make it conform to the XML schema.

2.4.5 Security of Policy Transport

P3P policies and references to P3P policies SHOULD NOT contain any sensitive information. This means that there are no additional security requirements for transporting a reference to a P3P policy beyond the requirements of the document it is associated with; so, if an HTML document would normally be served over a non-encrypted session, then P3P does **not** require nor recommend that the document be served over an encrypted session when a reference to a P3P policy is included with that document.

2.4.6 Policy Updates

Note that when a Web site changes its P3P policy, the old policy applies to data collected when it was in effect. It is the responsibility of the site to keep records of past P3P policies and policy reference files along with the dates when they were in effect, and to apply these policies appropriately.

If a site wishes to apply a new P3P policy to previously collected data, it MUST provide appropriate notice and opportunities for users to accept the new policy that are consistent with applicable laws, industry guidelines, or other privacy-related agreements the site has made.

2.4.7 Absence of Policy Reference File

If no policy reference file is available for a given site, user agents MUST assume (an empty) policy reference file exists at the well-known location with a 24 hour expiry, and therefore if the user returns to the site after 24 hours, the user agent MUST attempt to fetch a policy reference file from the well-known location again. User agents MAY check the well-known location more frequently, or upon a certain event such as the user clicking a browser refresh button. Sites MAY place a policy reference file at the well-known location that indicates that no policy is available, but set the expiry such that user agents know they need not check every 24 hours.

2.4.8 Asynchronous Evaluation

User agents MAY asynchronously fetch and evaluate P3P policies. That is, P3P policies need not necessarily be fetched and evaluated prior to other HTTP transactions.This behavior may be dependent on the the user's preferences and the type of request being made. Until a policy is evaluated, the user agent SHOULD treat the site as if it has no privacy policy. Once the policy has been evaluated, the user agent SHOULD apply the user's preferences. To promote deterministic behavior, the user agent SHOULD defer application of a policy until a consistent point in time. For example, a Web browser might apply a user's preferences just after the user agent completes a navigation, or when confirming a form submission.

2.5 Example Scenarios

As an aid to sites deploying P3P, several example scenarios are presented, along with descriptions of how P3P is used on those sites.

Scenario 1: Web site basic.example.com uses a variety of images, all of which it hosts. It also includes some forms, which are all submitted directly to the site. This site can declare a single P3P policy for the entire site (or if different privacy policies apply to different parts of the site, it can declare multiple P3P policies). As long as all of the images and form action URIs are in directories covered by the site's P3P policy, user agents will automatically recognize the images and forms as covered by the site's policy.

Scenario 2: Web site busy.example.com uses a content distribution network called cdn.example.com to host its images so as to reduce the load on its servers. Thus, all of the images on the site have URIs at cdn.example.com. CDN acts as an agent to Busy in this situation, and collects no data other than log data. This log data is used only for Web site and system administration in support of providing the services that Busy contracted for. Busy's privacy policy applies to the images hosted by CDN, so Busy uses the `HINT` element in its policy reference file to point to a suitable policy reference file at CDN, indicating that such images are covered by example.com P3P policy.

Scenario 3: Web site busy.example.com also has a contract with an advertising company called clickads.example.com to provide banner ads on its site. The contract allows Clickads to set cookies so as to make sure each user does not see a given ad more than three times. Clickads collects statistics on how many users view each ad and reports them to the companies whose products are being advertised. But these reports do not reveal information about any individual users. As was the case in Scenario 2, Busy's privacy policies applies to these ads hosted by Clickads, so Busy uses the `HINT` element in its policy reference file to point to a suitable policy reference file at Clickads, indicating that Busy P3P policy applies to such embedded content served by clickads.example.com. The companies whose products are being advertised need not be mentioned in the Busy privacy policy because the only data they are receiving is aggregate data.

Scenario 4: Web site busy.example.com also has a contract with funchat.example.com to host a chat room for its users. When users enter the chat room they are actually leaving the Busy site. However, the chat room has the Busy logo and is actually covered by the Busy privacy policy. In this instance Funchat is acting as an agent for Busy, but -- unlike the previous examples -- their content is not embedded in the Busy site. Busy can use the `HINT` element in its policy reference file to point to a suitable Funchat policy reference file, that indicates that Funchat chat room is covered by Busy privacy policy, therefore facilitating a smoother transition to the chat room.

Scenario 5: Web site bigsearch.example.com has a form that allows users to type in a search query and have it performed on their choice of search engines located on other sites. When a user clicks the "submit" button, the search query is actually submitted directly to these search engines -- the action URI is not on bigsearch.example.com but rather on the search engine selected by the user. Bigsearch can declare the privacy policies for these search engines by using the `HINT` element to point to their corresponding policy reference files. So when a user clicks the "submit" button, their user agent can check its privacy policy before posting any data. In order to make this search choice mechanism work, Bigsearch might actually have a form with an action URI on its own site, which redirects to the appropriate search engine. In this case, the user agent should check the search engine privacy policy upon receiving the redirect response.

Scenario 6: Web site bigsearch.example.com also has a form that allows users to type in a search query and have it simultaneously performed on ten different search engines. Bigsearch submits the queries, gets back the results from each search engine, removes the duplicates, and presents the results to the user. In this case, the user interacts only with Bigsearch. Thus, the only P3P policy involved is the one that covers the Bigsearch Web site. However, Bigsearch must disclose that it shares the users' search queries with third parties (the search Web sites), unless Bigsearch has a contract with these search engines and they act as agents to Bigsearch.

Scenario 7: Web site bigsearch.example.com also has banner advertisements provided by a company called adnetwork.example.com. Adnetwork uses cookies to develop profiles of users across many different Web sites so that it can provide them with ads better suited to their interests. Because the data about the sites that users are visiting is being used for purposes other than just serving ads on the Bigsearch Web site, Adnetwork cannot be considered an agent in this context. Adnetwork must create its own P3P policy and use its own policy reference file to indicate what content it applies to. In addition, Bigsearch may optionally use the `HINT` element in its policy reference file to indicate that the Adnetwork P3P policy reference file applies to these advertisements. Bigsearch should only do this if Adnetwork has told it what P3P policy applies to these advertisements and has agreed to notify Bigsearch if the policy reference needs to be changed.

Scenario 8: Web site busy.example.com uses cookies throughout its Web site. It discloses a cookie policy, separate from its regular P3P policy to cover these cookies. It uses the `COOKIE-INCLUDE` element in its policy reference file to declare the appropriate policy for these cookies. As a performance optimization, it also makes available a compact policy by sending a P3P header that includes this compact policy whenever it sets a cookie.

Scenario 9: Web site config.example.com provides a service in which they optimize various kinds of Web content based on each user's computer and Internet configuration. Users go to the Config Web site and answer questions about their computer, monitor, and Internet connection. Config encodes the responses and stores them in a cookie. Later, when the user is visiting Busy -- a Web site that has contracted with Config -- whenever the browser requests content that can be optimized (certain images, audio files, etc.), Busy will redirect the user to Config, which will read the user's cookie, and deliver the appropriate content. In this case, Config should declare a privacy policy that describes the kinds of data collected and stored in its cookies, and how that data is used. It should use a `COOKIE-INCLUDE` element in its policy reference file to declare the policy for the cookies. It will probably reference Busy's P3P policy for the actual images or audio files delivered, as it is acting much like CDN acts in scenario 2. Busy will probably also use `HINT` elements in its policy reference file to reference the policy for the Config-delivered content.

3. Policy Syntax and Semantics

P3P policies are encoded in XML with namespaces (see [XML] and [XML-Name]). A possible encoding using the RDF data model ([RDF]) is provided in [P3P-RDF].

Section 3.1 begins with an example of an English language privacy policy and a corresponding P3P policy. P3P policies include general assertions that apply to the entire policy as well as specific assertions -- called *statements* -- that apply only to the handling of particular types of data referred to by *data references*. Section 3.2 describes the `POLICY` element and policy-level assertions. Section 3.3 describes statements and data references.

3.1 Example policies

3.1.1 English language policies

The following are two examples of English-language privacy policy to be encoded as a P3P policy. Both policies are for one example company, CatalogExample, which has different policies for those browsing their site and those actually purchasing products. Example 3.1. is provided in both English and as a more formal description using P3P element and attribute names.

Example 3.1: CatalogExample's Privacy Policy for Browsers

> At CatalogExample, we care about your privacy. When you come to our site to look for an item, we will only use this information to improve our site and will not store it with information we could use to identify you.
>
> CatalogExample, Inc. is a licensee of the PrivacySealExample Program. The PrivacySealExample Program ensures your privacy by holding Web site licensees to high privacy standards and confirming with independent auditors that these information practices are being followed.
>
> Questions regarding this statement should be directed to:
> `CatalogExample`
> `4000 Lincoln Ave.`
> `Birmingham, MI 48009 USA`
> `email: catalog@example.com`
> `Telephone 248-EXAMPLE (248-392-6753)`
>
> If we have not responded to your inquiry or your inquiry has not been satisfactorily addressed, you can contact PrivacySealExample at http://www.privacyseal.example.org. CatalogExample will correct all errors or wrongful actions arising in connection with the privacy policy.
>
> *What We Collect and Why:*
> When you browse through our site we collect:
>
> - the basic information about your computer and connection to make sure that we can get you the proper information and for security purposes.
> - aggregate information on what pages consumers access or visit to improve our site.
>
> *Data retention:*
> We purge every two weeks the browsing information that we collect.

Here is Example 3.1 in a more formal description, using the P3P element and attribute names [with the section of the spec that was used cited in brackets for easy reference]:

- Disclosure URI: http://www.catalog.example.com/PrivacyPracticeBrowsing.html
  [3.2.2 Policy]
- Entity: CatalogExample
  4000 Lincoln Ave.
  Birmingham, MI 48009
  USA
  catalog@example.com
  +1 (248) 392-6753
  [3.2.4 Entity]

- Access to Identifiable Information: None
  [3.2.5 Access]
- Disputes:
  resolution type: independent
  service: http://www.privacyseal.example.org
  description: PrivacySealExample
  [3.2.6 Disputes]
- Remedies: we'll correct any harm done wrong
  [3.2.7 Remedies]
- We collect:
  dynamic.clickstream
  dynamic.http
  [4.5 Base data schema]
- For purpose: Web site and system administration, research and development
  [3.3.4 Purpose]
- Recipients: Only ourselves and our agents
  [3.3.5 Recipients]
- Retention: As long as appropriate for the stated purposes
  [3.3.6 Retention]
  (Note also that the site's human-readable privacy policy MUST mention that data is purged every two weeks, or provide a link to this information.)

Example 3.2: CatalogExample's Privacy Policy for Shoppers

At CatalogExample, we care about your privacy. We will never share your credit card number or any other financial information with any third party. With your permission only, we will share information with carefully selected marketing partners that meet either the preferences that you've specifically provided or your past purchasing habits. The more we and know about your likes and dislikes, the better we can tailor offerings to your needs.

CatalogExample is a licensee of the PrivacySealExample Program. The PrivacySealExample Program ensures your privacy by holding Web site licensees to high privacy standards and confirming with independent auditors that these information practices are being followed.

Questions regarding this statement should be directed to:

```
CatalogExample
4000 Lincoln Ave.
Birmingham, MI 48009 USA
email: catalog@example.com
Telephone +1 248-EXAMPLE (+1 248-392-6753)
```

If we have not responded to your inquiry or your inquiry has not been satisfactorily addressed, you can contact PrivacySealExample - http://privacyseal.example.org/privacyseal. CatalogExample will correct all errors or wrongful actions arising in connection with the privacy policy.

When you browse through our site we collect:

- the basic information about your computer and connection to make sure that we can get you the proper information and for security purposes; and
- aggregate information on what pages consumers access or visit to improve our site

If you choose to purchase an item we will ask you for more information including:

- your name and address so that we can have your purchase delivered to you and so we can contact you in the future;
- your email address and telephone number so we can contact you;
- a login and password to use to update your information at any time in the future; and
- financial information to complete your purchase (you may choose to store this for future use)
- optionally, you can enter other demographic information so that we can tailor services to you in the future.

Also on this page we will give you the option to choose if you would like to receive email, telephone calls or written service from CatalogExample or from our carefully selected marketing partners who maintain similar privacy practices. If you would like to receive these solicitations simply check the appropriate boxes. You can choose to stop participating at any time simply by changing your preferences.

Changing and Updating personal information
Consumers can change all of their personal account information by going to the preferences section of CatalogExample at http://catalog.example.com/preferences.html. You can change your address, telephone number, email address, password as well as your privacy settings.

Cookies
CatalogExample uses cookies only to see if you have been an CatalogExample customer in the past and, if so, customize services based on your past browsing habits and purchases. We do not store any personal data in the cookie nor do we share or sell the any of the information with other parties or affiliates.

Data retention
We will keep the information about you and your purchases for as long as you remain our customer. If you do not place an order from us for one year we will remove your information from our databases.

3.1.2 XML encoding of policies

The following pieces of [XML] capture the information as expressed in the above two examples. P3P policies are statements that are properly expressed as well-formed XML. The policy syntax will be explained in more detail in the sections that follow.

XML Encoding of Example 3.1:

```
<POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1">
 <POLICY name="forBrowsers"
     discuri="http://www.catalog.example.com/PrivacyPracticeBrowsing.html"
     xml:lang="en">
  <ENTITY>
   <DATA-GROUP>
    <DATA ref="#business.name">CatalogExample</DATA>
    <DATA ref="#business.contact-info.postal.street">4000 Lincoln Ave.</DATA>
    <DATA ref="#business.contact-info.postal.city">Birmingham</DATA>
    <DATA ref="#business.contact-info.postal.stateprov">MI</DATA>
    <DATA ref="#business.contact-info.postal.postalcode">48009</DATA>
    <DATA ref="#business.contact-info.postal.country">USA</DATA>
    <DATA ref="#business.contact-info.online.email">catalog@example.com</DATA>
    <DATA ref="#business.contact-info.telecom.telephone.intcode">1</DATA>
    <DATA ref="#business.contact-info.telecom.telephone.loccode">248</DATA>
    <DATA ref="#business.contact-info.telecom.telephone.number">3926753</DATA>
   </DATA-GROUP>
  </ENTITY>
  <ACCESS><nonident/></ACCESS>
  <DISPUTES-GROUP>
   <DISPUTES resolution-type="independent"
     service="http://www.PrivacySeal.example.org"
     short-description="PrivacySeal.example.org">
    <IMG src="http://www.PrivacySeal.example.org/Logo.gif" alt="PrivacySeal's logo"/>
    <REMEDIES><correct/></REMEDIES>
   </DISPUTES>
  </DISPUTES-GROUP>
  <STATEMENT>
   <PURPOSE><admin/><develop/></PURPOSE>
   <RECIPIENT><ours/></RECIPIENT>
   <RETENTION><stated-purpose/></RETENTION> <!-- Note also that the site's human-readable
                                                 privacy policy MUST mention that data
                                                 is purged every two weeks, or provide a
                                                 link to this information. -->
   <DATA-GROUP>
    <DATA ref="#dynamic.clickstream"/>
    <DATA ref="#dynamic.http"/>
   </DATA-GROUP>
  </STATEMENT>
 </POLICY>
</POLICIES>
```

XML Encoding of Example 3.2:

```
<POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1">
 <POLICY name="forShoppers"
     discuri="http://www.catalog.example.com/Privacy/PrivacyPracticeShopping.html"
     opturi="http://catalog.example.com/preferences.html"
     xml:lang="en">
  <ENTITY>
   <DATA-GROUP>
    <DATA ref="#business.name">CatalogExample</DATA>
    <DATA ref="#business.contact-info.postal.street">4000 Lincoln Ave.</DATA>
    <DATA ref="#business.contact-info.postal.city">Birmingham</DATA>
    <DATA ref="#business.contact-info.postal.stateprov">MI</DATA>
    <DATA ref="#business.contact-info.postal.postalcode">48009</DATA>
    <DATA ref="#business.contact-info.postal.country">USA</DATA>
    <DATA ref="#business.contact-info.online.email">catalog@example.com</DATA>
    <DATA ref="#business.contact-info.telecom.telephone.intcode">1</DATA>
    <DATA ref="#business.contact-info.telecom.telephone.loccode">248</DATA>
    <DATA ref="#business.contact-info.telecom.telephone.number">3926753</DATA>
   </DATA-GROUP>
  </ENTITY>
  <ACCESS><contact-and-other/></ACCESS>
  <DISPUTES-GROUP>
   <DISPUTES resolution-type="independent"
     service="http://www.PrivacySeal.example.org"
     short-description="PrivacySeal.example.org">
    <IMG src="http://www.PrivacySeal.example.org/Logo.gif" alt="PrivacySeal's logo"/>
    <REMEDIES><correct/></REMEDIES>
   </DISPUTES>
  </DISPUTES-GROUP>
  <STATEMENT>
   <CONSEQUENCE>
     We record some information in order to serve your request
     and to secure and improve our Web site.
   </CONSEQUENCE>
   <PURPOSE><admin/><develop/></PURPOSE>
   <RECIPIENT><ours/></RECIPIENT>
   <RETENTION><stated-purpose/></RETENTION>
   <DATA-GROUP>
    <DATA ref="#dynamic.clickstream"/>
    <DATA ref="#dynamic.http.useragent"/>
   </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>
   <CONSEQUENCE>
     We use this information when you make a purchase.
   </CONSEQUENCE>
   <PURPOSE><current/></PURPOSE>
   <RECIPIENT><ours/></RECIPIENT>
   <RETENTION><stated-purpose/></RETENTION>
   <DATA-GROUP>
    <DATA ref="#user.name"/>
    <DATA ref="#user.home-info.postal"/>
    <DATA ref="#user.home-info.telecom.telephone"/>
    <DATA ref="#user.business-info.postal"/>
    <DATA ref="#user.business-info.telecom.telephone"/>
    <DATA ref="#user.home-info.online.email"/>
    <DATA ref="#user.login.id"/>
    <DATA ref="#user.login.password"/>
    <DATA ref="#dynamic.miscdata">
```

```
     <CATEGORIES><purchase/></CATEGORIES>
    </DATA>
   </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>
   <CONSEQUENCE>
     At your request, we will send you carefully selected marketing
     solicitations that we think you will be interested in.
   </CONSEQUENCE>
   <PURPOSE>
    <contact required="opt-in"/>
    <individual-decision required="opt-in"/>
    <tailoring required="opt-in"/>
   </PURPOSE>
   <RECIPIENT><ours/><same required="opt-in"/></RECIPIENT>
   <RETENTION><stated-purpose/></RETENTION>
   <DATA-GROUP>
    <DATA ref="#user.name" optional="yes"/>
    <DATA ref="#user.home-info.postal" optional="yes"/>
    <DATA ref="#user.home-info.telecom.telephone" optional="yes"/>
    <DATA ref="#user.business-info.postal" optional="yes"/>
    <DATA ref="#user.business-info.telecom.telephone" optional="yes"/>
    <DATA ref="#user.home-info.online.email" optional="yes"/>
   </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>
   <CONSEQUENCE>
     We allow you to set a password so that you
     can access your own information.
   </CONSEQUENCE>
   <PURPOSE><individual-decision required="opt-in"/></PURPOSE>
   <RECIPIENT><ours/></RECIPIENT>
   <RETENTION><stated-purpose/></RETENTION>
   <DATA-GROUP>
    <DATA ref="#user.login.id"/>
    <DATA ref="#user.login.password"/>
     <CATEGORIES><uniqueid/></CATEGORIES>
    </DATA>
   </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>
   <CONSEQUENCE>
     At your request, we will tailor our site and
     highlight products related to your interests.
   </CONSEQUENCE>
   <PURPOSE>
     <pseudo-decision required="opt-in"/>
     <tailoring required="opt-in"/>
   </PURPOSE>
   <RECIPIENT><ours/></RECIPIENT>
   <RETENTION><stated-purpose/></RETENTION>
   <DATA-GROUP>
    <DATA ref="#user.bdate.ymd.year" optional="yes"/>
    <DATA ref="#user.gender" optional="yes"/>
   </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>
   <CONSEQUENCE>
     We tailor our site based on your past visits.
   </CONSEQUENCE>
   <PURPOSE><tailoring/><develop/></PURPOSE>
   <RECIPIENT><ours/></RECIPIENT>
   <RETENTION><stated-purpose/></RETENTION>
   <DATA-GROUP>
    <DATA ref="#dynamic.cookies">
     <CATEGORIES><state/></CATEGORIES>
    </DATA>
    <DATA ref="#dynamic.miscdata">
     <CATEGORIES><preference/></CATEGORIES>
    </DATA>
   </DATA-GROUP>
  </STATEMENT>
 </POLICY>
</POLICIES>
```

3.2 Policies

This section defines the syntax and semantics of P3P policies. All policies MUST be encoded using [UTF-8].

In cases where the P3P vocabulary is not precise enough to describe a Web site's practices, sites should use the vocabulary terms that most closely match their practices and provide further explanation in the `CONSEQUENCE` field and/or their human-readable policy. However, policies MUST NOT make false or misleading statements.

Policies have to be placed inside a `POLICIES` element.

3.2.1 The **`POLICIES`** element

The `POLICIES` element gathers one or more P3P policies together in a single file. This is provided as a performance optimization: many policies can be collected with a single request, improving network traffic and caching.

A `POLICIES` element is the root element of policy files. Further, the `POLICIES` element can be put within the policy reference file, inside the `META` element:: in this case, user agents need only fetch a single file, containing both the policy reference file and the policies.

The `POLICIES` element can optionally contain an `xml:lang` attribute (see section 2.4.2), an `EXPIRY` element, indicating the expiration of the included policies, and an embedded data schema using the `DATASCHEMA` element (see Section 5).

Since policies are included in a `POLICIES` element, each MUST have a `name` attribute which is unique in the file. This allows policy references (in `POLICY-REF` elements) to link to that policy.

Example 3.3:

The file in `http://www.example.com/Shop/policies.xml` could have the following content:

```
<POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1">
   <POLICY name="policy1" discuri="http://www.example.com/disc1"> .... </POLICY>
   <POLICY name="policy2" discuri="http://www.example.com/disc2"> .... </POLICY>
   <POLICY name="policy3" discuri="http://www.example.com/disc3"> .... </POLICY>
</POLICIES>
```

The files in `http://www.example.com/Shop/CDs/*` could then be associated to the second policy ("`policy2`") using the following policy reference file in `http://www.example.com/w3c/p3p.xml` :

```
<META xmlns="http://www.w3.org/2002/01/P3Pv1">
<POLICY-REFERENCES>
    <POLICY-REF about="/Shop/policies#policy2">
      <INCLUDE>/Shops/CDs/*</INCLUDE>
    </POLICY-REF>
 </POLICY-REFERENCES>
</META>
```

```
[19]    policies         =       `<POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1"` [xml-lang] `>`
                                 [expiry]
                                 [dataschema]
                                 *policy
                                 "</POLICIES>"
```

3.2.2 The **`POLICY`** element

The `POLICY` element contains a complete P3P policy. Each P3P policy MUST contain exactly one `POLICY` element. The policy element MUST contain an `ENTITY` element that identifies the legal entity making the representation of the privacy practices contained in the policy. In addition, the policy element MUST contain an `ACCESS` element, one or more `STATEMENT` elements, a `DISPUTES-GROUP` element, a P3P data schema, and one or more extensions.

`<POLICY>`
includes one or more statements. Each statement includes a set of disclosures as applied to a set of data elements.

`name` **(*mandatory attribute*)**
name of the policy, used as a fragment identifier to be able to reference the policy.

`discuri` **(*mandatory attribute*)**
URI of the natural language privacy statement.

`opturi`
URI of instructions that users can follow to request or decline to have their data used for a particular purpose (opt-in or opt-out). This attribute is ***mandatory*** for policies that contain a `purpose` with required attribute set to `opt-in` or `opt-out`. Note that the opt-in or opt-out procedures are determined by each site and may not necessarily include a central mechanism for the entire site or an automated online mechanism.

`xml:lang`
Language in which the policy is expressed (see section 2.4.2).

```
[20]    policy                    =          `<POLICY name=` quotedstring
                                                     ` discuri=` quoted-URI
                                                     [` opturi=` quoted-URI]
                                                     [xml-lang] `>`
                                             *extension
                                             [test]
                                             entity
                                             access
                                             [disputes-group]
                                             1*statement-block
                                             *extension
                                             `</POLICY>`
[21]    quoted-URI                =          `"` URI `"`
```

Here, URI is defined as per RFC 2396 [URI].

3.2.3 The **`TEST`** element

The TEST element is used for testing purposes: the presence of `TEST` in a policy indicates that the policy is just an example, and as such, it MUST be ignored, and not be considered as a valid P3P policy.

```
[22]            test                        =                      "<TEST/>"
```

3.2.4 The **`ENTITY`** element

The `ENTITY` element gives a precise description of the legal entity making the representation of the privacy practices.

`<ENTITY>`
identifies the legal entity making the representation of the privacy practices contained in the policy

The `ENTITY` element contains a description of the legal entity consisting of DATA elements referencing (all or some of) the fields of the business dataset: it MUST contain both the legal entity's name and one or more contact information fields among postal address, telephone number, email address, URI. Note that some laws and codes of conduct require entities to include a postal address or other specific information in their contact information.

```
[23]   entity                    =    "<ENTITY>"
                                      *extension
                                      entitydescription
                                      *extension
                                      "</ENTITY>"

[24]   entitydescription         =    "<DATA-GROUP>"
                                      `<DATA ref="#business.name"/>` PCDATA "</DATA>"
                                      *(`<DATA ref="#business.` string `"/>` PCDATA "</DATA>")
                                      "</DATA-GROUP>"
```

Here, `string` is defined as a sequence of characters (with " and & escaped) among the values that are allowed by the business dataset. `PCDATA` is defined as in [XML].

3.2.5 The `ACCESS` element

The `ACCESS` element indicates whether the site provides access to various kinds of information.

`<ACCESS>`

the ability of the individual to view identified data and address questions or concerns to the service provider. Service providers MUST disclose one value for the access attribute. The method of access is not specified. Any disclosure (other than `<all/>`) is not meant to imply that access to all data is possible, but that some of the data may be accessible and that the user should communicate further with the service provider to determine what capabilities they have.

Note that service providers may also wish to provide capabilities to access information collected through means other than the Web at the **discuri.** However, the scope of P3P statements are limited to data collected through HTTP or other Web transport protocols. Also, if access is provided through the Web, use of strong authentication and security mechanisms for such access is recommended; however, security issues are outside the scope of this document.

The `ACCESS` element must contain one of the following elements:

`<nonident/>`

Web site does not collect identified data.

`<all/>`

All Identified Data: access is given to all identified data.

`<contact-and-other/>`

Identified Contact Information and Other Identified Data: access is given to identified online and physical contact information as well as to certain other identified data.

`<ident-contact/>`

Identifiable Contact Information: access is given to identified online and physical contact information (e.g., users can access things such as a postal address).

`<other-ident/>`

Other Identified Data: access is given to certain other identified data (e.g., users can access things such as their online account charges).

`<none/>`

None: no access to identified data is given.

```
[25]   access                    =    "<ACCESS>"
                                      *extension
                                      access_disclosure
                                      *extension
                                      "</ACCESS>"

[26]   access_disclosure         =    "<nonident/>"           | ; Identified Data is Not Used


                                      "<all/>"                | ; All Identifiable Information
                                      "<contact-and-other/>"  | ; Identified Contact Information and
                                                                  Other Identified Data
                                      "<ident-contact/>"      | ; Identifiable Contact Information
                                      "<other-ident/>"        | ; Other Identified Data
                                      "<none/>"                 ; None
```

3.2.6 The `DISPUTES` element

A policy SHOULD contain a `DISPUTES-GROUP` element, which contains one or more `DISPUTES` elements. These elements describe dispute resolution procedures that may be followed for disputes about a services' privacy practices. Each `DISPUTES` element can optionally contain a `LONG-DESCRIPTION` element, an `IMG` element, and a `REMEDIES` element. Service providers with multiple dispute resolution procedures should use a separate `DISPUTES` element for each. Since different dispute procedures have separate remedy processes, each `DISPUTES` element would need a separate `LONG-DESCRIPTION`, `IMG` tag and `REMEDIES` element, if they are being used.

`<DISPUTES>`

Describes dispute resolution procedures that may be followed for disputes about a services' privacy practices, or in case of protocol violation.

`resolution-type` (*mandatory attribute*)

takes one of the following four values:

Customer Service `[service]`

Individual may complain to the Web site's customer service representative for resolution of disputes regarding the use of collected data. The description MUST include information about how to contact customer service.

Independent Organization `[independent]`

Individual may complain to an independent organization for resolution of disputes regarding the use of collected data. The description MUST include information about how to contact the third party organization.

Court `[court]`

Individual may file a legal complaint against the Web site.

Applicable Law `[law]`

Disputes arising in connection with the privacy statement will be resolved in accordance with the law referenced in the description.

service (***mandatory attribute***)
: URI of the customer service Web page or independent organization, or URI for information about the relevant court or applicable law

verification
: URI or certificate that can be used for verification purposes. It is anticipated that seal providers will provide a mechanism for verifying a site's claim that they have a seal.

short-description
: A short human readable description of the name of the appropriate legal forum, applicable law, or third party organization; or contact information for customer service if not already provided at the service URI. No more than 255 characters.

The DISPUTES element can contain a LONG-DESCRIPTION element, where a human readable description is present: this should contain the name of the appropriate legal forum, applicable law, or third party organization; or contact information for customer service if not already provided at the service URI.

<LONG-DESCRIPTION>
: This element contains a (possibly long) human readable description.

<IMG>
: An image logo (for example, of the independent organization or relevant court)

src (***mandatory attribute***)
: URI of the image logo

width
: width in pixels of the image logo

height
: height in pixels of the image logo

alt (***mandatory attribute***)
: very short textual alternative for the image logo

```
[27]    disputes-group         =      "<DISPUTES-GROUP>"
                                      *extension
                                      1*dispute
                                      *extension
                                      "</DISPUTES-GROUP>"

[28]    dispute                =      "<DISPUTES"
                                       " resolution-type=" '"'("service"|"independent"|"court"|"law")'"'
                                       " service=" quoted-URI
                                       [" verification=" quotedstring]
                                       [" short-description=" quotedstring]
                                      ">"
                                      *extension
                                      [longdescription]
                                      [image]
                                      [remedies]
                                      *extension
                                      "</DISPUTES>"

[29]    longdescription        =      <LONG-DESCRIPTION> PCDATA </LONG-DESCRIPTION>

[30]    image                  =      "<IMG src=" quoted-URI
                                      [" width=" `"` number `"`]
                                      [" height=" `"` number `"`]
                                      " alt=" quotedstring
                                      "/>"

[31]    quotedstring           =      `"` string `"`
```

Here, string is defined as a sequence of characters (with " and & escaped), and PCDATA is defined as in [XML].

Note that there can be multiple assurance services, specified via multiple occurrences of DISPUTES within the DISPUTES-GROUP element. These fields are expected to be used in a number of ways, including representing that one's privacy practices are self assured, audited by a third party, or under the jurisdiction of a regulatory authority.

3.2.7 The REMEDIES element

Each DISPUTES element SHOULD contain a REMEDIES element that specifies the possible remedies in case a policy breach occurs.

<REMEDIES>
: Remedies in case a policy breach occurs.

The REMEDIES element must contain one or more of the following:

<correct/>
: Errors or wrongful actions arising in connection with the privacy policy will be remedied by the service.

<money/>
: If the service provider violates its privacy policy it will pay the individual an amount specified in the human readable privacy policy or the amount of damages.

<law/>
: Remedies for breaches of the policy statement will be determined based on the law referenced in the human readable description.

```
[32]    remedies                  =         "<REMEDIES>"
                                            *extension
                                            1*remedy
                                            *extension
                                            "</REMEDIES>"

[33]    remedy                    =         "<correct/>" |
                                            "<money/>"   |
                                            "<law/>"
```

3.3 Statements

Statements describe data practices that are applied to particular types of data.

3.3.1 The STATEMENT element

The STATEMENT element is a container that groups together a PURPOSE element, a RECIPIENT element, a RETENTION element, a DATA-GROUP element, and optionally a CONSEQUENCE element and one or more extensions. All of the data referenced by the DATA-GROUP is handled according to the disclosures made in the other elements contained by the statement. Thus, sites may group elements that are handled the same way and create a statement for each group. Sites that would prefer to disclose separate purposes and other information for each kind of data they collect can do so by creating a separate statement for each data element.

<STATEMENT>
data practices as applied to data elements.

```
[34]      statement-block         =      "<STATEMENT>"
                                         *extension
                                         [consequence]
                                         ((purpose recipient retention 1*data-group) |
                                          (non-identifiable [purpose] [recipient] [retention] *data-group))
                                         *extension
                                         "</STATEMENT>"
```

To simplify practice declaration, service providers may aggregate any of the disclosures (purposes, recipients, and retention) within a statement over data elements. Service providers MUST make such aggregations as an additive operation. For instance, a site that distributes your age to ours (ourselves and our agents), but distributes your postal code to unrelated (unrelated third parties), MAY say they distribute your name and postal code to ours and unrelated. Such a statement appears to distribute more data than actually happens. It is up to the service provider to determine if their disclosure deserves specificity or brevity. Note that when aggregating disclosures across statements that include the NON-IDENTIFIABLE element, this element may be included in the aggregated statement only if it would otherwise appear in every statement if the statements were written separately.

Also, one must always disclose all options that apply. Consider a site with the sole purpose of collecting information for the purposes of contact (Contacting Visitors for Marketing of Services or Products). Even though this is considered to be for the current (Completion and Support of Activity For Which Data Was Provided) purpose, the site must state both contact and current purposes. Consider a site which distributes information to ours in order to redistribute it to public: the site must state both ours and public recipients.

Service providers often aggregate data they collect. Sometimes this aggregate data may be used for different purposes than the original data, shared more widely than the original data, or retained longer than the original data. For example many sites publish or disclose to their advertisers statistics such as number of visitors to their Web site, percentage of visitors who fit into various demographic groups, etc. When aggregate statistics are used or shared such that it would not be possible to derive data for individual people or households based on these statistics, no disclosures about these statistics are necessary in a P3P policy. However, services MUST disclose the fact that the original data is collected and declare any use that is made of the data before it is aggregated.

3.3.2 The CONSEQUENCE element

STATEMENT elements may optionally contain a CONSEQUENCE element that can be shown to a human user to provide further explanation about a site's practices.

<CONSEQUENCE>
Consequences that can be shown to a human user to explain why the suggested practice may be valuable in a particular instance even if the user would not normally allow the practice.

```
[35]           consequence                             =               "<CONSEQUENCE>"
                                                                       PCDATA
                                                                       "</CONSEQUENCE>"
```

3.3.3 The NON-IDENTIFIABLE element

A STATEMENT element may optionally contain the NON-IDENTIFIABLE element, signifying either that there is no data collected under this STATEMENT, or that all of the data referenced by that STATEMENT will be anonymized upon collection.

<NON-IDENTIFIABLE/>
This element signifies that either no data is collected (including Web logs), or that the organization collecting the data will anonymize the data referenced in the enclosing STATEMENT. In order to consider the data "anonymized", there must be no reasonable way for the entity or a third party to attach the collected data to the identity of a natural person. Some types of data are inherently anonymous, such as randomly-generated session IDs. Data which might identify natural people in some circumstances, such as IP addresses, names, or addresses, must have a non-reversible transformation applied in order be considered "anonymized".
An example of a non-reversible transformation is removing the last seven bits of an IP address and replacing them with zeros. This transformation must be applied to all copies of the data, including those that might be stored on backup media. An algorithm that replaces identified data with unique corresponding values from a table is not considered non-reversible. In addition, a one-way cryptographic hash would not be considered non-reversible if the set of possible data values is small enough that all possible hashed values can be generated and compared with the value that someone is attempting to reverse.

If the NON-IDENTIFIABLE element is present in any STATEMENT elements in a policy, then a human readable explanation of how the data is anonymized MUST be included or linked to at the **discuri** .

Also, if the NON-IDENTIFIABLE element is present in a STATEMENT then the other elements in that STATEMENT are optional.

```
[36]          non-identifiable                           =            "<NON-IDENTIFIABLE/>"
```

3.3.4 The PURPOSE element

Each STATEMENT element that does not include a NON-IDENTIFIABLE element MUST contain a PURPOSE element that contains one or more purposes of data collection or uses of data. Sites MUST classify their data practices into one or more of the purposes specified below.

`<PURPOSE>`
purposes for data processing relevant to the Web.

The `PURPOSE` element MUST contain one or more of the following:

`<current/>`
Completion and Support of Activity For Which Data Was Provided: Information may be used by the service provider to complete the activity for which it was provided, whether a one-time activity such as returning the results from a Web search, forwarding an email message, or placing an order; or a recurring activity such as providing a subscription service, or allowing access to an online address book or electronic wallet.

`<admin/>`
Web Site and System Administration: Information may be used for the technical support of the Web site and its computer system. This would include processing computer account information, information used in the course of securing and maintaining the site, and verification of Web site activity by the site or its agents.

`<develop/>`
Research and Development: Information may be used to enhance, evaluate, or otherwise review the site, service, product, or market. This does not include personal information used to tailor or modify the content to the specific individual nor information used to evaluate, target, profile or contact the individual.

`<tailoring/>`
One-time Tailoring: Information may be used to tailor or modify content or design of the site where the information is used only for a single visit to the site and not used for any kind of future customization. For example, an online store might suggest other items a visitor may wish to purchase based on the items he has already placed in his shopping basket.

`<pseudo-analysis/>`
Pseudonymous Analysis: Information may be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals *for purpose of research, analysis and reporting*, but it will not be used to attempt to identify specific individuals. For example, a marketer may wish to understand the interests of visitors to different portions of a Web site.

`<pseudo-decision/>`
Pseudonymous Decision: Information may be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals *to make a decision that directly affects that individual*, but it will not be used to attempt to identify specific individuals. For example, a marketer may tailor or modify content displayed to the browser based on pages viewed during previous visits.

`<individual-analysis/>`
Individual Analysis: Information may be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data *for the purpose of research, analysis and reporting*. For example, an online Web site for a physical store may wish to analyze how online shoppers make offline purchases.

`<individual-decision/>`
Individual Decision: Information may be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data *to make a decision that directly affects that individual*. For example, an online store suggests items a visitor may wish to purchase based on items he has purchased during previous visits to the Web site.

`<contact/>`
Contacting Visitors for Marketing of Services or Products: Information may be used to contact the individual, through a communications channel other than voice telephone, for the promotion of a product or service. This includes notifying visitors about updates to the Web site. This does not include a direct reply to a question or comment or customer service for a single transaction -- in those cases, **`<current/>`** would be used. In addition, this does not include marketing via customized Web content or banner advertisements embedded in sites the user is visiting -- these cases would be covered by the **`<tailoring/>`**, **`<pseudo-analysis/>`** and **`<pseudo-decision/>`**, or **`<individual-analysis/>`** and **`<individual-decision/>`** purposes.

`<historical/>`
Historical Preservation: Information may be archived or stored for the purpose of preserving social history as governed by an existing law or policy. This law or policy MUST be referenced in the `<DISPUTES>` element and MUST include a specific definition of the type of qualified researcher who can access the information, where this information will be stored and specifically how this collection advances the preservation of history.

`<telemarketing/>`
Contacting Visitors for Marketing of Services or Products Via Telephone: Information may be used to contact the individual via a voice telephone call for promotion of a product or service. This does not include a direct reply to a question or comment or customer service for a single transaction -- in those cases, **`<current/>`** would be used.

`<other-purpose>` ***string*** **`</other-purpose>`**
Other Uses: Information may be used in other ways not captured by the above definitions. (A human readable explanation MUST be provided in these instances).

Each type of purpose (with the exception of `current`) can have the following optional attribute:

`required`
Whether the purpose is a required practice for the site. The attribute can take the following values:

- **`always`** : The purpose is always required; users cannot opt-in or opt-out of this use of their data. This is the default when no `required` attribute is present.
- **`opt-in`** : Data may be used for this purpose only when the user affirmatively requests this use -- for example, when a user asks to be added to a mailing list. An affirmative request requires users to take some action specifically to make the request. For example, when users fill out a survey, checking an additional box to request to be added to a mailing list would be considered an affirmative request. However, submitting a survey form that contains a pre-checked mailing list request box would not be considered an affirmative request. In addition, for any purpose that users may affirmatively request, there must also be a way for them to change their minds later and decline -- this MUST be specified at the `opturi`.
- **`opt-out`** : Data may be used for this purpose unless the user requests that it not be used in this way. When this value is selected, the service MUST provide clear instructions to users on how to opt-out of this purpose at the `opturi`. Services SHOULD also provide these instructions or a pointer to these instructions at the point of data collection.

[37]
```
purpose          =    "<PURPOSE>"
                      *extension
                      1*purposevalue
                      *extension
                      "</PURPOSE>"
```

[38]
```
purposevalue     =    "<current/>"                  | ; Completion and Support of Activity For Which Data
                      "<admin" [required]   "/>"    | ; Web Site and System Administration
                      "<develop" [required] "/>"    | ; Research and Development
                      "<tailoring" [required] "/>"           | ; One-time Tailoring
                      "<pseudo-analysis" [required] "/>"     | ; Pseudonymous Analysis
                      "<pseudo-decision" [required] "/>"     | ; Pseudonymous Decision
                      "<individual-analysis" [required] "/>" | ; Individual Analysis
                      "<individual-decision" [required] "/>" | ; Individual Decision
                      "<contact" [required] "/>"             | ; Contacting Visitors for Marketing of Services or P
                      "<historical" [required] "/>"          | ; Historical Preservation
                      "<telemarketing" [required] "/>"       | ; Telephone Marketing
                      "<other-purpose" [required] ">" PCDATA "</other-purpose>"; Other Uses
[39]  required      =  " required=" `"` ("always"|"opt-in"|"opt-out") `"`
```

Service providers MUST use the above elements to explain the purpose of data collection. Service providers MUST disclose *all that apply*. If a service provider does not disclose that a data element will be used for a given purpose, that is a representation that data will not be used for that purpose. Service providers that disclose that they use data for "`other`" purposes MUST provide human readable explanations of those purposes.

3.3.5 The **`RECIPIENT`** element

Each `STATEMENT` element that does not include a `NON-IDENTIFIABLE` element MUST contain a `RECIPIENT` element that contains one or more recipients of the collected data. Sites MUST classify their recipients into one or more of the six recipients specified.

`<RECIPIENT>`
the legal entity, or domain, beyond the service provider and its agents where data may be distributed.

The `RECIPIENT` element MUST contain one or more of the following:

`<ours>`
Ourselves and/or entities acting as our agents or entities for whom we are acting as an agent: An agent in this instance is defined as a third party that processes data only on behalf of the service provider for the completion of the stated purposes. (e.g., the service provider and its printing bureau which prints address labels and does nothing further with the information.)

`<delivery>`
Delivery services possibly following different practices: Legal entities *performing delivery services* that may use data for purposes other than completion of the stated purpose. This should also be used for delivery services whose data practices are unknown.

`<same>`
Legal entities following our practices: Legal entities who use the data on their own behalf under equable practices. (e.g., consider a service provider that grants the user access to collected personal information, and also provides it to a partner who uses it once but discards it. Since the recipient, who has otherwise similar practices, cannot grant the user access to information that it discarded, they are considered to have equable practices.)

`<other-recipient>`
Legal entities following different practices: Legal entities that are constrained by and accountable to the original service provider, but may use the data in a way not specified in the service provider's practices (e.g., the service provider collects data that is shared with a partner who may use it for other purposes. However, it is in the service provider's interest to ensure that the data is not used in a way that would be considered abusive to the users' and its own interests.)

`<unrelated>`
Unrelated third parties: Legal entities whose data usage practices are not known by the original service provider.

`<public>`
Public fora: Public fora such as bulletin boards, public directories, or commercial CD-ROM directories.

Each of the above tags can optionally contain:

- one or more `recipient-description` tags, containing a description of the recipient;
- with the exception of `<ours>`, a `required` attribute: this attribute is defined exactly as the analogous attribute in the `PURPOSE` tag, indicating whether opt-in/opt-out of sharing is available (and, its default value is `always`).

```
[40]  recipient       =   "<RECIPIENT>"
                          *extension
                          1*recipientvalue
                          *extension
                          "</RECIPIENT>"

[41]  recipientvalue  =   "<ours>" *recdescr
                          "</ours>"                         |  ; only ourselves and our agents
                          "<same" [required] ">" *recdescr
                          "</same>"                         |  ; legal entities following our practices
                          "<other-recipient" [required] ">" *recdescr
                          "</other-recipient>"              |  ; legal entities following different practices
                          "<delivery" [required] ">" *recdescr
                          "</delivery>"                     |  ; delivery services following different practices
                          "<public" [required] ">" *recdescr
                          "</public>"                       |  ; public fora
                          "<unrelated" [required] ">" *recdescr
                          "</unrelated>"                       ; unrelated third parties

[42]  recdescr        =   "<recipient-description>"
                          PCDATA                               ; description of the recipient
                          "</recipient-description>"
```

Service providers MUST disclose *all the recipients that apply*. P3P makes no distinctions about how that data is released to the recipient; it simply requires that if data is released, then that sharing must be disclosed in the P3P policy. Examples of disclosing data which MUST be covered by a P3P statement include:

- Transmitting customer data as part of an order-fulfillment or billing process
- Leasing or selling mailing lists
- Placing personal information in URIs when redirecting requests to a third party
- Placing personal information in URIs which link to a third party

Note that in some cases the above set of recipients may not completely describe all the recipients of data. For example, the issue of transaction facilitators, such as shipping or payment processors, who are necessary for the completion and support of the activity but may follow different practices was problematic. Currently, only delivery services can be explicitly represented in a policy. Other such transaction facilitators should be represented in whichever category most accurately reflects their practices with respect to the original service provider.

A special element for delivery services is included, but not one for payment processors (such as banks or credit card companies) for the following reasons: Financial institutions will typically have separate agreements with their customers regarding the use of their financial data, while delivery recipients typically do not have an opportunity to review a delivery service's privacy policy.

Note that the `<delivery/>` element SHOULD NOT be used for delivery services that agree to use data only on behalf of the service provider for completion of the delivery.

3.3.6 The **`RETENTION`** element

Each `STATEMENT` element that does not include a `NON-IDENTIFIABLE` element MUST contain a `RETENTION` element that indicates the kind of retention policy that applies to the data referenced in that statement.

`<RETENTION>`

the type of retention policy in effect

The `RETENTION` element MUST contain one of the following:

`<no-retention/>`

Information is not retained for more than a brief period of time necessary to make use of it during the course of a single online interaction. Information MUST be destroyed following this interaction and MUST NOT be logged, archived, or otherwise stored. This type of retention policy would apply, for example, to services that keep no Web server logs, set cookies only for use during a single session, or collect information to perform a search but do not keep logs of searches performed.

`<stated-purpose/>`

For the stated purpose: Information is retained to meet the stated purpose. This requires information to be discarded at the earliest time possible. Sites MUST have a retention policy that establishes a destruction time table. The retention policy MUST be included in or linked from the site's human-readable privacy policy.

`<legal-requirement/>`

As required by law or liability under applicable law: Information is retained to meet a stated purpose, but the retention period is longer because of a legal requirement or liability. For example, a law may allow consumers to dispute transactions for a certain time period; therefore a business may for liability reasons decide to maintain records of transactions, or a law may affirmatively require a certain business to maintain records for auditing or other soundness purposes. Sites MUST have a retention policy that establishes a destruction time table. The retention policy MUST be included in or linked from the site's human-readable privacy policy.

`<business-practices/>`

Determined by service provider's business practice: Information is retained under a service provider's stated business practices. Sites MUST have a retention policy that establishes a destruction time table. The retention policy MUST be included in or linked from the site's human-readable privacy policy.

`<indefinitely/>`

Indefinitely: Information is retained for an indeterminate period of time. The absence of a retention policy would be reflected under this option. Where the recipient is a public fora, this is the appropriate retention policy.

```
[43]   retention              =       "<RETENTION>"
                                      *extension
                                      retentionvalue
                                      *extension
                                      "</RETENTION>"

[44]   retentionvalue         =       "<no-retention/>"       | ; not retained
                                      "<stated-purpose/>"     | ; for the stated purpose
                                      "<legal-requirement/>"  | ; stated purpose by law
                                      "<indefinitely/>"       | ; indeterminate period of time
                                      "<business-practices/>"   ; by business practices
```

3.3.7 The **`DATA-GROUP`** and **`DATA`** elements

Each `STATEMENT` element that does not include a `NON-IDENTIFIABLE` element MUST contain at least one `DATA-GROUP` element that contains one or more `DATA` elements. `DATA` elements are used to describe the type of data that a site collects.

`<DATA-GROUP>`

describes the data to be transferred or inferred

`base`

base URI ([URI]) for URI references present in `ref` attributes. When this attribute is omitted, the default value is the URI of the P3P base data schema (http://www.w3.org/TR/P3P/base). When the attribute appears as an empty string (""), the base is the local document.

`<DATA>`

describes the data to be transferred or inferred

`ref` ***(mandatory attribute)***

URI reference ([URI]), where the fragment identifier part denotes the *name of a data element/set*, and the URI part denotes the corresponding *data schema*. In case the URI part is not present, if the `DATA` element is contained within a `DATA-GROUP` element, then the default base URI is assumed to be the URI of the `base` attribute. In the other cases, as usual, the default base URI is a same-document reference ([URI]).

Remember that ***names of data elements and sets are case-sensitive*** (so, for example, `user.gender` is different from `USER.GENDER` or `User.Gender`).

`optional`

indicates whether or not the site requires visitors to submit this data element to access a resource or complete a transaction; "no" indicates that the data element is not optional (it is required), while "yes" indicates that the data element is optional. *The default is "no."* The `optional` attribute is used only in policies (not in data schema definitions).

Note that user agents should be cautious about using the `optional` attribute in automated decision-making. If the `optional` attribute is associated with a data element directly controlled by the user agent (such as the HTTP `Referer` header or cookies), the user agent should make sure that this data is not transmitted to Web sites at which a data element is optional if the site's policy would not match a user's preferences if the data element was required. Likewise, for data elements that users typically type into forms, user agents should alert users when a site's practices about optional data do not match their preferences.

DATA elements can contain the actual data (as already sen in the case of the ENTITY element), and can contain related category information.

```
[45]     data-group          =      "<DATA-GROUP"
                                    [" base=" quoted-URI]
                                    ">"
                                    *extension
                                    1*dataref
                                    *extension
                                    "</DATA-GROUP>"

[46]     dataref             =      `<DATA" ref="` URI-reference `"`
                                     [" optional=" `"` ("yes"|"no") `"`] ">"
                                     [categories] ; the categories of the data element.
                                     [PCDATA] ; the eventual value of the data element
                                    "</DATA>"
```

Here, URI-reference is defined as in [URI].

For example, to reference the user's home address city, all the elements of the data set user.business-info and (optionally) all the elements of the data set user.home-info.telecom, the service would send the following references inside a P3P policy:

```
<DATA-GROUP>
<DATA ref="#user.home-info.city"/>
<DATA ref="#user.home-info.telecom" optional="yes"/>
<DATA ref="#user.business-info"/>
</DATA-GROUP>
```

When the actual value of the data is known, it can be expressed inside the DATA element. For example, as seen in the example policies:

```
 <ENTITY>
  <DATA-GROUP>
   <DATA ref="#business.name">CatalogExample</DATA>
   <DATA ref="#business.contact-info.postal.street">4000 Lincoln Ave.</DATA>
...
```

3.4 Categories and the CATEGORIES element

Categories are elements inside data elements that provide hints to users and user agents as to the intended uses of the data. Categories are vital to making P3P user agents easier to implement and use. Note that *categories are not data elements*: they just allow users to express more generalized preferences and rules over the exchange of their data. Categories SHOULD NOT be used within the DATA elements of an ENTITY element.

The following elements are used to denote data categories:

```
[47]    categories          =      "<CATEGORIES>" 1*category "</CATEGORIES>"

[48]    category            =      "<physical/>"    | ; Physical Contact Information
                                   "<online/>"      | ; Online Contact Information
                                   "<uniqueid/>"    | ; Unique Identifiers
                                   "<purchase/>"    | ; Purchase Information
                                   "<financial/>"   | ; Financial Information
                                   "<computer/>"    | ; Computer Information
                                   "<navigation/>"  | ; Navigation and Click-stream Data
                                   "<interactive/>" | ; Interactive Data
                                   "<demographic/>" | ; Demographic and Socioeconomic Data
                                   "<content/>"     | ; Content
                                   "<state/>"       | ; State Management Mechanisms
                                   "<political/>"   | ; Political Information
                                   "<health/>"      | ; Health Information
                                   "<preference/>"  | ; Preference Data
                                   "<location/>"    | ; Location Data
                                   "<government/>   | ; Government-issued Identifiers
                                   "<other-category>" PCDATA "</other-category>" ; Other
```

<physical/>

Physical Contact Information: Information that allows an individual to be contacted or located in the physical world -- such as telephone number or address.

<online/>

Online Contact Information: Information that allows an individual to be contacted or located on the Internet -- such as email. Often, this information is independent of the specific computer used to access the network. (See the category "Computer Information")

<uniqueid/>

Unique Identifiers: Non-financial identifiers, excluding government-issued identifiers, issued for purposes of consistently identifying or recognizing the individual. These include identifiers issued by a Web site or service.

<purchase/>

Purchase Information: Information actively generated by the purchase of a product or service, including information about the method of payment.

<financial/>

Financial Information: Information about an individual's finances including account status and activity information such as account balance, payment or overdraft history, and information about an individual's purchase or use of financial instruments including credit or debit card information. Information about a discrete purchase by an individual, as described in "Purchase Information," alone does not come under the definition of "Financial Information."

<computer/>

Computer Information: Information about the computer system that the individual is using to access the network -- such as the IP number, domain name, browser type or operating system.

<navigation/>

Navigation and Click-stream Data: Data *passively* generated by *browsing* the Web site -- such as which pages are visited, and how long users stay on each page.

<interactive/>

Interactive Data: Data *actively* generated from or reflecting *explicit interactions* with a service provider through its site -- such as queries to a search engine, or logs of account activity.

`<demographic/>`
: **Demographic and Socioeconomic Data**: Data about an individual's characteristics -- such as gender, age, and income.

`<content/>`
: **Content** : The words and expressions contained in the body of a communication -- such as the text of email, bulletin board postings, or chat room communications.

`<state/>`
: **State Management Mechanisms**: Mechanisms for maintaining a stateful session with a user or automatically recognizing users who have visited a particular site or accessed particular content previously -- such as HTTP cookies.

`<political/>`
: **Political Information**: Membership in or affiliation with groups such as religious organizations, trade unions, professional associations, political parties, etc.

`<health/>`
: **Health Information**: information about an individual's physical or mental health, sexual orientation, use or inquiry into health care services or products, and purchase of health care services or products.

`<preference/>`
: **Preference Data**: Data about an individual's likes and dislikes -- such as favorite color or musical tastes.

`<location/>`
: **Location Data**: Information that can be used to identify an individual's current physical location and track them as their location changes -- such as GPS position data.

`<government/>`
: **Government-issued Identifiers**: Identifiers issued by a government for purposes of consistently identifying the individual.

`<other-category>` ***string*** **`</other-category>`**
: **Other**: Other types of data not captured by the above definitions. (A human readable explanation should be provided in these instances, between the `<other-category>` and the `</other-category>` tags.)

The **Computer**, **Navigation**, **Interactive** and **Content** categories can be distinguished as follows. The Computer category includes information about the user's computer including IP address and software configuration. Navigation data describes actual user behavior related to browsing. When an IP address is stored in a log file with information related to browsing activity, both the Computer category and the Navigation category should be used. Interactive Data is data actively solicited to provide some useful service at a site beyond browsing. Content is information exchanged on a site for the purposes of communication.

The **Other** category should be used only when data is requested that does not fit into any other category.

P3P uses categories to give users and user agents additional hints as to what type of information is requested from a service. While most data in the base data schema is in a known category (or a set of known categories), some data elements can be in a number of different categories, depending on the situation. The former are called *fixed-category data elements* (or "fixed data elements" for short), the latter *variable-category data elements* ("variable data elements"). Both types of elements are described in Section 5.7.

3.5 Extension Mechanism: the `EXTENSION` element

P3P provides a flexible and powerful mechanism to extend its syntax and semantics using one element: `EXTENSION`. This element is used to indicate portions of the policy/policy reference file/data schema which belong to an extension. The meaning of the data within the `EXTENSION` element is defined by the extension itself.

`<EXTENSION>`
: describes an extension to the syntax

`optional`
: This attribute determines if the extension is *mandatory* or *optional*. A *mandatory* extension is indicated by giving the `optional` attribute a value of `no`. A *mandatory* extension to the P3P syntax means that applications that do not understand this extension cannot understand the meaning of the whole policy (or policy reference file, or data schema) containing it. An *optional* extension, indicated by giving the optional attribute a value of `yes`, means that applications that do not understand this extension can safely ignore the contents of the `EXTENSION` element, and proceed to process the whole policy (or policy reference file, or data schema) as usual. The `optional` attribute is not required; its default value is `yes`.

[49]
```
extension        =      "<EXTENSION" [" optional=" `"` ("yes"|"no") `"`] ">" PCDATA "</EXTENSION>"
```

For example, if www.catalog.example.com would like to add to P3P a feature to indicate that a certain set of data elements were only to be collected from users living in the United States, Canada, or Mexico, it could add a mandatory extension like this:

```
<DATA-GROUP>
...
<EXTENSION optional="no">
<COLLECTION-GEOGRAPHY type="include" xmlns="http://www.catalog.example.com/P3P/region">
<USA/><Canada/><Mexico/>
</COLLECTION-GEOGRAPHY>
</EXTENSION>
</DATA-GROUP>
```

On the other hand, if www.catalog.example.com would like to add an extension stating what country the server is in, an optional extension might be more appropriate, such as the following:

```
<POLICY>
<EXTENSION optional="yes">
<ORIGIN xmlns="http://www.catalog.example.com/P3P/origin" country="USA"/>
</EXTENSION>
...
</POLICY>
```

The `xmlns` attribute is significant since it specifies the namespace for interpreting the names of elements and attributes used in the extension. Note that, as specified in [XML-Name], the namespace URI is just intended to be a unique identifier for the XML entities used by the extension. Nevertheless, service providers MAY provide a page with a description of the extension at the corresponding URI.

The `EXTENSION` element can appear in various places within P3P syntax: such positions are normatively specified by the XML Schema present in Appendix 4 (and, informally specified by the ABNF syntax, and by the DTD present in Appendix 5)..

3.6 User Preferences

User agents MUST document a method by which preferences can be imported and processed, and SHOULD document a method by which preferences can be exported.

P3P user agents MUST act according to the preference settings selected by the user. This requires that they be able to process policy and policy reference files as appropriate to evaluate each policy with respect to a user's preferences or other criteria specified by the settings. Depending on these settings, this may require, for example, that the user agent verify that required parts of the P3P policy are present, or check that the syntax of the entire policy is valid.

4. Compact Policies

Compact policies are summarized P3P policies that provide hints to user agents to enable the user agent to make quick, synchronous decisions about applying policy. Compact policies are a performance optimization that is **OPTIONAL** for either user agents or servers. User agents that are unable to obtain enough information from a compact policy to make a decision according to a user's preferences SHOULD fetch the full policy.

In P3P, compact policies contain policy information related to cookies (cf. [COOKIES] and [STATE]) only. The Web server is responsible for building a P3P compact policy to represent the cookies referenced in a full policy. The policy specified in a P3P compact policy applies to data stored within all cookies set in the same HTTP response as the compact policy, all cookies set by scripts associated with that HTTP response, and also to data linked to the cookies.

4.1 Referencing compact policies

Any HTTP resource MAY include a P3P compact policy through the P3P response header (cf. Section 2.2.2). If a site is using P3P headers, it SHOULD include this on responses for all appropriate request methods, including `HEAD` and `OPTION` requests.

The P3P compact policy header has a quoted string that may contain one or more delimited tokens (the "compact policy"). Tokens can appear in any order, and the space character (" ") is the only valid delimiter. The syntax for this header is as follows:

```
[50]  compact-policy-field     =    `CP="` compact-policy `"`
[51]  compact-policy           =    compact-token *(" " compact-token)
[52]  compact-token            =    compact-access            |
                                    compact-disputes          |
                                    compact-remedies          |
                                    compact-non-identifiable  |
                                    compact-purpose           |
                                    compact-recipient         |
                                    compact-retention         |
                                    compact-categories        |
                                    compact-test
```

As for all HTTP headers, the name of the P3P header field is case-insensitive. The field-value (i.e., the content of the header) is instead case sensitive.

If an HTTP response includes more than one compact policy, P3P user agents MUST ignore all compact policies after the first one.

4.2 Compact Policy Vocabulary

P3P compact policies use tokens representing the following elements from the P3P vocabulary: `ACCESS`, `CATEGORIES`, `DISPUTES`, `NON-INDENTIFIABLE`, `PURPOSE`, `RECIPIENT`, `REMEDIES`, `RETENTION`, `TEST`.

If a token appears more than once in a single compact policy, the compact policy has *the same semantics* as if that token appeared only once. If an unrecognized token appears in a compact policy, the compact policy has *the same semantics* as if that token was not present.

The P3P compact policy vocabulary is expressed using a developer-readable language to reduce the number of bytes transferred over the wire within a HTTP response header. The syntax of the tokens follows:

4.2.1 Compact ACCESS

Information in the `ACCESS` element is represented in compact policies using tokens composed by a three letter code:

```
[53]  compact-access    =    "NOI" | ; for <nonident/>
                             "ALL" | ; for <all/>
                             "CAO" | ; for <contact-and-other/>
                             "IDC" | ; for <ident-contact/>
                             "OTI" | ; for <other-ident/>
                             "NON"   ; for <none/>
```

4.2.2 Compact DISPUTES

If a full P3P policy contains a `DISPUTES-GROUP` element that contains one or more `DISPUTES` elements, then the server should signal the user agent by providing a **single** "`DSP`" token in the P3P-compact policy field:

```
[54]  compact-disputes    =    "DSP" ; there are some DISPUTES
```

4.2.3 Compact REMEDIES

Information in the `REMEDIES` element is represented in compact policies as follows:

```
[55]  compact-remedies    =    "COR" | ; for <correct/>
```

```
                                              "MON" | ; for <money/>
                                              "LAW"   ; for <law/>
```

4.2.4 Compact `NON-IDENTIFIABLE`

The presence of the `NON-IDENTIFIABLE` element in every statement of the policy is signaled by the `NID` token (note that the `NID` token MUST NOT be used unless the `NON-IDENTIFIABLE` element is present in every statement within the policy):

```
[56]  compact-non-identifiable   =   "NID" ; for <NON-IDENTIFIABLE/>
```

4.2.5 Compact `PURPOSE`

Purposes are expressed in P3P compact policy format using tokens composed by a three letter code plus an optional one letter attribute. Such an optional attribute encodes the value of the "`required`" attribute in full P3P policies: its value can be "`a`", "`i`" and "`o`", which mean that the "`required`" attribute in the corresponding P3P policy must be set to "`always`", "`opt-in`" and "`opt-out`" respectively.

If a P3P compact policy needs to specify one or more other-purposes in its full P3P policy, a single `OTP` flag is used to signal the user agent that other-purposes exist in the full P3P policy.

The corresponding associations among P3P purposes and compact policy codes follow:

```
[57]  compact-purpose   =   "CUR"        | ; for <current/>
                            "ADM" [creq] | ; for <admin/>
                            "DEV" [creq] | ; for <develop/>
                            "TAI" [creq] | ; for <tailoring/>
                            "PSA" [creq] | ; for <pseudo-analysis/>
                            "PSD" [creq] | ; for <pseudo-decision/>
                            "IVA" [creq] | ; for <individual-analysis/>
                            "IVD" [creq] | ; for <individual-decision/>
                            "CON" [creq] | ; for <contact/>
                            "HIS" [creq] | ; for <historical/>
                            "TEL" [creq] | ; for <telemarketing/>
                            "OTP" [creq]   ; for <other-purpose/>
```

```
[58]  creq              =   "a"| ;"always"
                            "i"| ;"opt-in"
                            "o"  ;"opt-out"
```

4.2.6 Compact `RECIPIENT`

Recipients are expressed in P3P compact policy format using a three letter code plus an optional one letter attribute. Such an optional attribute encodes the value of the "`required`" attribute in full P3P policies: its value can be "`a`", "`i`" and "`o`", which mean that the "`required`" attribute in the corresponding P3P policy must be set to "`always`", "`opt-in`" and "`opt-out`" respectively.

The corresponding associations among P3P recipients and compact policy codes follow:

```
[59]  compact-recipient   =   "OUR"        | ; for <ours/>
                              "DEL" [creq] | ; for <delivery/>
                              "SAM" [creq] | ; for <same/>
                              "UNR" [creq] | ; for <unrelated/>
                              "PUB" [creq] | ; for <public/>
                              "OTR" [creq]   ; for <other-recipient/>
```

4.2.7 Compact `RETENTION`

Information in the `RETENTION` element is represented in compact policies as follows:

```
[60]  compact-retention   =   "NOR" | ; for <no-retention/>
                              "STP" | ; for <stated-purpose/>
                              "LEG" | ; for <legal-requirement/>
                              "BUS" | ; for <business-practices/>
                              "IND"   ; for <indefinitely/>
```

4.2.8 Compact `CATEGORIES`

Categories are represented in compact policies as follows:

```
[61]  compact-categories   =   "PHY" | ; for <physical/>
                               "ONL" | ; for <online/>
                               "UNI" | ; for <uniqueid/>
                               "PUR" | ; for <purchase/>
                               "FIN" | ; for <financial/>
                               "COM" | ; for <computer/>
                               "NAV" | ; for <navigation/>
                               "INT" | ; for <interactive/>
                               "DEM" | ; for <demographic/>
                               "CNT" | ; for <content/>
                               "STA" | ; for <state/>
                               "POL" | ; for <political/>
                               "HEA" | ; for <health/>
                               "PRE" | ; for <preference/>
                               "LOC" | ; for <location/>
                               "GOV" | ; for <government/>
                               "OTC"   ; for <other-category/>
```

Note that if a P3P policy specifies one or more `other-category` in its full P3P policy, a **single** `OTC` token is used to signal the user agent that other-category's exist in the full P3P policy.

4.2.9 Compact TEST

The presence of the TEST element is signaled by the TST token:

```
[62]    compact-test                       =              "TST" ; for <TEST/>
```

4.3 Compact Policy Scope

When a P3P compact policy is included in a HTTP response header, it applies to cookies set by the current response. This includes cookies set through the use of a HTTP SET-COOKIE header or cookies set by script.

4.4 Compact Policy Lifetime

To use compact policies, the validity of the full P3P policy must span the lifetime of the cookie. There is no method to indicate that policy is valid beyond the life of the cookie because the value of user agent caching is marginal, since sites would not know when to optimize by not sending the compact policy. When a server sends a compact policy, it is asserting that the compact policy and corresponding full P3P policy will be in effect for at least the lifetime of the cookie to which it applies.

4.5 Transforming a P3P Policy to a Compact Policy

When using P3P compact policies, the Web site is responsible for building a compact policy by summarizing the policy referenced by the COOKIE-INCLUDE elements of a P3P policy reference file. If a site's policy reference file uses COOKIE-EXCLUDE elements then the site will need to manage sending the correct P3P compact policies to the user agent given the cookies set in a specific response.

The transformation of a P3P policy to a P3P compact policy may result in a loss of descriptive policy information -- the compact policy may not contain all of the policy information specified in the full P3P policy. The information from the full policy that is discarded when building a compact policy includes expiry, data group/data-schema elements, entity elements, consequences elements, and disputes elements are reduced.

Full policies that include mandatory extensions MUST NOT be represented as compact policies.

All of the purposes, recipients, and categories that appear in multiple statements in a full policy MUST be aggregated in a compact policy, as described in section 3.3.1. When performing the aggregation, a Web site MUST disclose all relevant tokens (for instance, observe Example 4.1, where multiple retention policies are specified.)

In addition, for each fixed category data element appearing in a statement the associated category as defined in the associated schema MUST be included in the compact policy.

Example 4.1:

Consider the following P3P policy:

```
<POLICY name="sample"
  discuri="http://www.example.com/cookiepolicy.html"
  opturi="http://www.example.com/opt.html">
  <ENTITY>
    <DATA-GROUP>
      <DATA ref="#business.name">Example, Corp.</DATA>
      <DATA ref="#business.contact-info.online.email">privacy@example.com</DATA>
    </DATA-GROUP>
  </ENTITY>
  <ACCESS><none/></ACCESS>
  <DISPUTES-GROUP>
    <DISPUTES resolution-type="service"
     service="http://www.example.com/privacy.html"
     short-description="Please contact our customer service desk with
                        privacy concerns by emailing privacy@example.com"/>
  </DISPUTES-GROUP>
  <STATEMENT>
    <PURPOSE><admin/><develop/><pseudo-decision/></PURPOSE>
    <RECIPIENT><ours/></RECIPIENT>
    <RETENTION><indefinitely/></RETENTION>
    <DATA-GROUP>
      <DATA ref="#dynamic.cookies">
        <CATEGORIES><preference/><navigation/></CATEGORIES>
      </DATA>
    </DATA-GROUP>
  </STATEMENT>
  <STATEMENT>
    <PURPOSE><individual-decision required="opt-out"/></PURPOSE>
    <RECIPIENT><ours/></RECIPIENT>
    <RETENTION><stated-purpose/></RETENTION>
    <DATA-GROUP>
      <DATA ref="#user.name.given"/>
      <DATA ref="#dynamic.cookies">
        <CATEGORIES><preference/><uniqueid/></CATEGORIES>
      </DATA>
    </DATA-GROUP>
  </STATEMENT>
</POLICY>
```

The corresponding compact policy is:

```
"NON DSP ADM DEV PSD IVDo OUR IND STP PHY PRE NAV UNI"
```

4.6 Transforming a Compact Policy to a P3P Policy

Some user agents may attempt to generate a full P3P policy from a compact policy, for use in evaluating user preferences. They will not be able to provide values for the ENTITY and DISPUTES elements as well as a number of the attributes. However:

In case there *are not* multiple different values of compact retention,
they should be able to generate a policy with an appropriate ACCESS element, and: a single STATEMENT element that contains the appropriate RECIPIENT, RETENTION, and PURPOSE elements, as well as a dynamic.miscdata element with the appropriate CATEGORIES.

In case there *are* multiple different values of compact retention,
they should be able to generate a policy with an appropriate ACCESS element, and: multiple STATEMENT elements (as many as the different values of the compact retention) that contain a different corresponding value for the RETENTION element, the appropriate RECIPIENT, and PURPOSE elements, as well as a dynamic.miscdata element with the appropriate CATEGORIES.

Note that, in agreement with the non ambiguity requirements stated in Section 2.4.1, a site MUST honor a compact policy for a given URI in any case (even when the full policy referenced in the policy reference file for that URI does not correspond, as per Section 4.5, to the compact policy itself).

5. Data Schemas

A *data schema* is a description of a set of data. P3P includes a way to describe data schemas so that services can communicate to user agents about the data they collect. A data schema is built from a number of *data elements*, which are specific items of data a service might collect.

Data elements in a data schema can have the following properties:

- Data element name. The name of the data element is used when a P3P policy includes this data element in a <DATA> element. This is required on all data elements.
- Descriptive name or short name. A data element's short name provides a short, human-understandable name for the data element. The short name is not required, but it is strongly recommended.
- Long description. The long description of a data element provides a more detailed, human-understandable definition of the data element. Like the short name, the long description is not required, but it is strongly recommended.
- Category or categories. Most data elements have categories assigned to them when they are defined in a data schema. See Categories for more information on categories.

Data elements are organized into a hierarchy. A data element automatically includes all of the data elements below it in the hierarchy. For example, the data element representing "the user's name" includes the data elements representing "the user's given name", "the user's family name", and so on. The hierarchy is based on the data element name. Thus the data elements user.name.given, user.name.family, and user.name.nickname are all children of the data element user.name, which is in turn a child of the data element user.

P3P has defined a data schema called the *P3P base data schema* that includes a large number of data elements commonly used by services.

Services may declare new data elements by creating and publishing their own data schemas, which are created using the <DATASCHEMA> element. Data schemas can either be published in standalone XML files (whose root element is then DATASCHEMA), or they can be embedded in a policy file (even the same policy file with policies referencing the data schema itself).

The <DATASCHEMA> element is defined as follows:

```
[63]   dataschema       =      "<DATASCHEMA" [` xmlns="http://www.w3.org/2002/01/P3Pv1"`] [xml-lang] ">"
                               *(datadef|datastruct|extension)
                               "</DATASCHEMA>"
```

A standalone data schema has the <DATASCHEMA> element as the root element in the XML file. It must have the appropriate namespace defined in the xmlns attribute to identify it as a P3P data schema, as follows:

```
<DATASCHEMA xmlns="http://www.w3.org/2002/01/P3Pv1">
<DATA-STRUCT ... />
...
<DATA-DEF ... />
</DATASCHEMA>
```

Optionally, DATASCHEMA can contain an xml:lang attribute (see section 2.4.2),

When a data schema is declared inside a policy file, then the <DATASCHEMA> element is still used (as described in Section 3.2.1, "The <POLICIES> element").

5.1 Natural Language Support for Data Schemas

Data schemas contain a number of fields in natural language. Services publishing a data schema MAY wish to translate these fields into multiple languages. The data element short and long names MAY be translated, but the data element name MUST NOT be translated - this field needs to stay constant across translations of a data schema.

If a service is going to provide a data schema in multiple natural languages, then it SHOULD examine the Accept-Language HTTP request-header on requests for that data schema to pick the best available alternative.

5.2 Data Structures

Data schemas often need to reuse a common group of data elements. P3P data schemas support this through data structures. A data structure is a named, abstract definition of a group of data elements. When a data element is defined, it can be defined as being of an unstructured type, in which case it has no child elements. The data element can also be defined as being of a specific structured type, in which case the data element will be automatically expanded to include as sub-elements all of the elements defined in the data structure. For example, the following structure is used to represent a date and time:

```
<!-- "date" Data Structure -->
<DATA-STRUCT name="date.ymd.year"
    short-description="Year"/>

<DATA-STRUCT name="date.ymd.month"
    short-description="Month"/>

<DATA-STRUCT name="date.ymd.day"
    short-description="Day"/>

<DATA-STRUCT name="date.hms.hour"
    short-description="Hour"/>

<DATA-STRUCT name="date.hms.minute"
    short-description="Minute"/>

<DATA-STRUCT name="date.hms.second"
    short-description="Second"/>
```

Now we shall define a "meeting" data element, which has a time and place for the meeting:

```
<DATA-DEF name="meeting.time"
    short-description="Meeting time"
    structref="#date"/>
<DATA-DEF name="meeting.place"
    short-description="Meeting place/>
```

Since `meeting.place` does not reference a structure, it is of an unstructured type, and has no child elements. The `meeting.time` element uses the `date` structure. By declaring this, the following sub-elements are created:

```
meeting.time.ymd.year
meeting.time.ymd.month
meeting.time.ymd.day
meeting.time.hms.hour
meeting.time.hms.minute
meeting.time.hms.second
```

A P3P policy can now declare that it collects the `meeting` data element, which implies that it collects all of the sub-elements of `meeting`, or it can use data elements lower down the hierarchy - `meeting.time`, for example, or `meeting.time.ymd.day`.

5.3 The `DATA-DEF` and `DATA-STRUCT` elements

`<DATA-DEF>` and `<DATA-STRUCT>`

Define a data element or a data structure, respectively. Data structures are reusable structured type definitions that can be used to build data elements. Data elements are declared within a `<STATEMENT>` in a P3P policy to describe data covered by that statement.

The following attributes are common to these two elements:

`name` ***(mandatory attribute)***

Indicates the name of the data element or data structure. Remember that names of data element and data structures are **case-sensitive**, so, for example, `user.gender` is different from `USER.GENDER` or `User.Gender`. Furthermore, in names of data elements and structures no number character can appear immediately following a dot.

`structref`

URI reference ([URI]), where the fragment identifier part denotes the *structure*, and the URI part denotes the corresponding *data schema* where it is defined. The default base URI is a same-document reference ([URI]). Data elements or data structures without a `structref` attribute (and, so, without an associated structure) are called *unstructured*.

`short-description`

a string denoting the short display name of the data element or structure, no more than 255 characters.

The `DATA-DEF` and `DATA-STRUCT` elements can also contain a long description of the data element or structure, using the `LONG-DESCRIPTION` element.

```
[64]   datadef        =    "<DATA-DEF name=" quotedstring
                            [` structref="` URI-reference `"`]
                            [" short-description=" quotedstring]
                            ">"
                            [categories] ; the categories of the data element.
                            [longdescription] ; the long description of the data element
                           "</DATA-DEF>"

[65]   datastruct     =    "<DATA-STRUCT name=" quotedstring
                            [` structref="` URI-reference `"`]
                            [" short-description=" quotedstring]
                            ">"
                            [categories] ; the categories of the Data Structure.
                            [longdescription] ; the long description of the Data Structure
                           "</DATA-STRUCT>"
```

Here, `URI-reference` is defined as in [URI].

Data elements can be structured, much like in common programming languages: structures are hierarchical (tree-like) descriptions of data elements: this hierarchical description is performed in the `name` attribute using a dot ("`.`") character as separator.

P3P provides the ***P3P base data schema***, which has built-in definitions of a number of widely used structures and data elements. All P3P implementations are required to understand the P3P base data schema, so the structures and elements it defines are always available to P3P implementers.

A data schema may include multiple `DATA-STRUCT` elements that together describe a structure. For example, there is no single `DATA-STRUCT` for the `uri` data structure (cf. section 5.5.7.1) in the P3P base data schema. Instead `uri.authority`, `uri.stem`, and `uri.querystring` are interpreted together to define this structure.

5.3.1 Categories in P3P Data Schemas

Categories can be assigned to data structures or data elements. The following rules define how those category definitions are meant to be used:

1. `<DATA-STRUCT>` elements MAY include category definitions. If a structure definition includes categories, then all uses of those structures in data definitions and data structures pick up those categories. If a structure contains no categories, then the categories for that structure MAY be defined when it is used in another structure or data element. Otherwise, a data element using this structure is a variable-category element. Any uses of a variable-category data element in a policy require that its categories be listed in the policy.
2. A `<DATA-DEF>` with an unstructured type is a variable-category data element if no categories are defined in the `<DATA-DEF>`, and has exactly those categories listed in the `<DATA-DEF>` if any categories are included.
3. A `<DATA-DEF>` or `<DATA-STRUCT>` with a structured type which has no categories defined on that structure produces a variable-category data element/structure if no categories are defined in the `<DATA-DEF>` or `<DATA-STRUCT>`. If the `<DATA-DEF>` or `<DATA-STRUCT>` does have categories listed, then those categories are applied to that data element, and all of its sub-elements. In other words, categories are pushed down into sub-elements when defining a data element to be of a structured type, and the structured type does not define any categories.
4. A `<DATA-DEF>` using a structured type which has categories defined on that structure picks up all the categories listed on the structure. In addition, categories may be listed in the `<DATA-DEF>`, and these are added to the categories defined in the structure. These categories are defined only at the level of that data element, and are not "pushed down" to any sub-elements.
5. A `<DATA-STRUCT>` that has no categories assigned to it, and which is using a structured subtype which has categories defined on the subtype picks up all the categories listed on the subtype.
6. A `<DATA-STRUCT>` that has categories assigned to it, and which is using a structured subtype replaces all of the categories listed on the subtype.
7. There is a "bubble-up" rule for categories when referencing data elements: data elements, must at a minimum, include all categories defined by any of its children. This rule applies recursively, so for example, all categories defined by data elements `foo.a.w`, `foo.a.y`, and `foo.b.z` MUST be considered to apply to data element `foo`.
8. A `<DATA-STRUCT>` cannot be defined with some variable-category elements and some fixed-category elements. Either all of the sub-elements of a structure must be in the variable category, or else all of them must have one or more assigned categories.
9. A `<DATA-DEF>` with some variable-category elements and some fixed-category elements MUST NOT be referenced. Note, this means that the `dynamic` structure (cf. section 5.6.4 "Dynamic Data"), existing in the basedata schema, cannot be referenced in a policy (each of its sub-elements `dynamic.clickstream`, `dynamic.http`, etc. can be referenced individually).

5.3.2 P3P Data Schema Example

Consider the case where the company HyperSpeedExample wishes to describe the features of a vehicle, using a structure called `vehicle`. This structure includes:

- The vehicle's model type (`vehicle.model`),
- The vehicle's color (`vehicle.color`),
- The vehicle's year of manufacture (`vehicle.built.year`), and
- The vehicle's price (`vehicle.price`).

If HyperSpeedExample also wants to include in the definition of a vehicle the location of manufacture, it could add other fields to the structure with all the relevant data like country, street address, postal code, and so on. But, each part of a structure can use other structures as well: *structures can be composed*. In this case, the ***P3P base data schema*** already provides a structure `postal`, describing all the postal information of a location. So, the final definition of the structure vehicle is

- `vehicle.model` (unstructured)
- `vehicle.color` (unstructured)
- `vehicle.price` (unstructured)
- `vehicle.built.year` (unstructured)
- `vehicle.built.where` (with structure `postal` from the base data schema)

The structure `postal` has fields `postal.street`, `postal.city`, and so on. Since we have applied the structure `postal` to `vehicle.built.where`, it means that we can access the street and city of a vehicle using the descriptions `vehicle.built.where.street` and `vehicle.built.where.city` respectively. So, by applying a structure (in this case, `postal`) we can build very complex descriptions in a modular way.

HyperSpeedExample wants to declare that all of the vehicle information will be in the `<preference/>` category. The `vehicle.model, vehicle.color, vehicle.price,` and `vehicle.built.year` fields are all unstructured types, so assigning them to the `<preference/>` category accomplishes this for those fields. Since vehicle is a structure definition, assigning the `<preference/>` category to `vehicle.built.where` will override (replace) the categories defined on all of the sub-elements of `vehicle.built.where`, placing all of them in the `<preference/>` category, even though the `postal` structure was originally defined as being in other categories.

As said, structures do not contain data elements; they are just abstract data types. We can use them to rapidly build structured collections of data elements. Going on with the example, HyperSpeedExample needs this abstract description of the features of a vehicle because it wants to actually exchange data about cars and motorcycles. So, it could define two data elements called `car` and `motorcycle`, both with the above structure `vehicle`.

This description of the data elements and data structures is encoded in XML using a data schema. In the HyperSpeedExample case, it would be something like:

```
<DATASCHEMA xmlns="http://www.w3.org/2002/01/P3Pv1">
<DATA-STRUCT name="vehicle.model"
    short-description="Model">
    <CATEGORIES><preference/></CATEGORIES>
</DATA-STRUCT>
<DATA-STRUCT name="vehicle.color"
    short-description="Color">
    <CATEGORIES><preference/></CATEGORIES>
</DATA-STRUCT>
<DATA-STRUCT name="vehicle.built.year"
    short-description="Construction Year">
```

```
        <CATEGORIES><preference/></CATEGORIES>
    </DATA-STRUCT>
    <DATA-STRUCT name="vehicle.built.where"
        structref="http://www.w3.org/TR/P3P/base#postal"
        short-description="Construction Place">
        <CATEGORIES><preference/></CATEGORIES>
    </DATA-STRUCT>
    <DATA-DEF name="car" structref="#vehicle"/>
    <DATA-DEF name="motorcycle" structref="#vehicle"/>
    </DATASCHEMA>
```

Continuing with the example, in order to reference a car model and construction year, HyperSpeedExample *or any other service* could send the following references inside a P3P policy:

```
    <DATA-GROUP>
      <!-- First, the "car.model" data element, whose definition is in the data schema
           at http://www.HyperSpeed.example.com/models-schema
        -->
    <DATA ref="http://www.HyperSpeed.example.com/models-schema#car.model"/>

      <!-- And second, the "car.built.year" data element, whose definition is the data schema
           at http://www.HyperSpeed.example.com/models-schema
        -->
    <DATA ref="http://www.HyperSpeed.example.com/models-schema#car.built.year"/>
    </DATA-GROUP>
```

Using the `base` attribute, the above references can be written in an even more compact way:

```
    <DATA-GROUP base="http://www.HyperSpeed.example.com/models-schema">
        <DATA ref="#car.model"/>
        <DATA ref="#car.built.year"/>
    </DATA-GROUP>
```

Alternatively, the data schema could be *embedded* directly into a policy file. In this case, the policy file could look like:

```
    <POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1">
    <!-- Embedded data schema -->
    <DATASCHEMA>
    <DATA-STRUCT name="vehicle.model"
        short-description="Model">
        <CATEGORIES><preference/></CATEGORIES>
    </DATA-STRUCT>
    <DATA-STRUCT name="vehicle.color"
        short-description="Color">
        <CATEGORIES><preference/></CATEGORIES>
    </DATA-STRUCT>
    <DATA-STRUCT name="vehicle.built.year"
        short-description="Construction Year"">
        <CATEGORIES><preference/></CATEGORIES>
    </DATA-STRUCT>
    <DATA-STRUCT name="vehicle.built.where"
        structref="http://www.w3.org/TR/P3P/base#postal"
        short-description="Construction Place">
        <CATEGORIES><preference/></CATEGORIES>
    </DATA-STRUCT>
    <DATA-DEF name="car" structref="#vehicle"/>
    <DATA-DEF name="motorcycle" structref="#vehicle"/>
    </DATASCHEMA>
    <!-- end of embedded data schema -->
    <POLICY name="policy1" discuri="http://www.example.com/disc1">
    ...
    <DATA-GROUP base="">
    <DATA ref="#car.model"/>
    <DATA ref="#car.built.year"/>
    </DATA-GROUP>
    ...
    </POLICY>
    <POLICY name="policy2" discuri="http://www.example.com/disc2"> .... </POLICY>
    <POLICY name="policy3" discuri="http://www.example.com/disc3"> .... </POLICY>
    </POLICIES>
```

Note that in any case there MUST NOT be more than one data schema per file.

5.3.3 Use of data element names

Note that the data element names specified in the base data schema or in extension data schemas may be used for purposes other than P3P policies. For example, Web sites may use these names to label HTML form fields. By referring to data the same way in P3P policies and forms, automated form-filling tools can be better integrated with P3P user agents.

5.4 Persistence of data schemas

An essential requirement on data schemas is the ***persistence of data schemas***: data schemas that can be fetched at a certain URI can only be changed by extending the data schema in a *backward-compatible* way (that is to say, changing the data schema does not change the meaning of any policy using that schema). This way, the URI of a policy acts in a sense like a unique identifier for the data elements and structures contained therein: any data schema that is not backward-compatible *must therefore use a new different URI*.

Note that a useful application of the persistence of data schema is given for example in the case of multi-lingual sites: multiple language versions (translations) of the same data schema can be offered by the server, using the HTTP "`Content-Language`" response header field to properly indicate that a particular language has been used for the data schema.

5.5 Basic Data Structures

The Basic Data Structures are structures used by the P3P base data schema (and possibly, due to their basic nature, they should be reused as much as possible by other different data schemas). All P3P-compliant user agent implementations MUST be aware of the Basic Data Structures. Each table below specifies the elements of a basic data structure, the categories associated, their structures, and the display names shown to users. More than one category may be associated with a fixed data element. However, each base data element is assigned to only one category whenever possible. Data schema designers are recommended to do the same.

5.5.1 Dates

The **date** structure specifies a date. Since date information can be used in different ways, depending on the context, all **date** information is tagged as being of "variable" category (see Section 5.7.2). For example, schema definitions can explicitly set the corresponding category in the element referencing this data structure, where soliciting the birthday of a user might be "Demographic and Socioeconomic Data", while the expiration date of a credit card might belong to the "Purchase Information" category.

`date`	**Category**	**Structure**	**Short display name**
ymd.year	*(variable-category)*	*unstructured*	Year
ymd.month	*(variable-category)*	*unstructured*	Month
ymd.day	*(variable-category)*	*unstructured*	Day
hms.hour	*(variable-category)*	*unstructured*	Hour
hms.minute	*(variable-category)*	*unstructured*	Minute
hms.second	*(variable-category)*	*unstructured*	Second
fractionsecond	*(variable-category)*	*unstructured*	Fraction of Second
timezone	*(variable-category)*	*unstructured*	Time Zone

The "time zone" information is for example described in the time standard [ISO8601]. Note that "date.ymd" and "date.hms" can be used to fast reference the year/month/day and hour/minute/second blocks respectively.

5.5.2 Names

The **personname** structure specifies information about the naming of a person.

`personname`	**Category**	**Structure**	**Short display name**
prefix	Demographic and Socioeconomic Data	*unstructured*	Name Prefix
given	Physical Contact Information	*unstructured*	Given Name (First Name)
family	Physical Contact Information	*unstructured*	Family Name (Last Name)
middle	Physical Contact Information	*unstructured*	Middle Name
suffix	Demographic and Socioeconomic Data	*unstructured*	Name Suffix
nickname	Demographic and Socioeconomic Data	*unstructured*	Nickname

5.5.3 Logins

The **login** structure specify information (IDs and passwords) for computer systems and Web sites which require authentication. Note that this data element should not be used for computer systems or Web sites which use digital certificates for authentication: in those cases, the *certificate* structure should be used.

`login`	**Category**	**Structure**	**Short display name**
id	Unique Identifiers	*unstructured*	Login ID
password	Unique Identifiers	*unstructured*	Login Password

The "id" field represents the ID portion of the login information for a computer system. Often, user IDs are made public, while passwords are kept secret. IDs do not include any type of biometric authentication mechanisms.

The "password" field represents the password portion of the login information for a computer system. This is a secret data value, usually a character string, that is used in authenticating a user. Passwords are typically kept secret, and are generally considered to be sensitive information

5.5.4 Certificates

The **certificate** structure is used to specify identity certificates (like, for example, X.509).

`certificate`	**Category**	**Structure**	**Short display name**
key	Unique Identifiers	*unstructured*	Certificate Key
format	Unique Identifiers	*unstructured*	Certificate Format

The "format" field is used to represent the information of an IANA registered public key or authentication certificate format, while the "key" field is used to represent the corresponding certificate key.

5.5.5 Telephones

The **telephonenum** structure specifies the characteristics of a telephone number.

`telephonenum`	**Category**	**Structure**	**Short display name**

intcode	Physical Contact Information	*unstructured*	International Telephone Code
loccode	Physical Contact Information	*unstructured*	Local Telephone Area Code
number	Physical Contact Information	*unstructured*	Telephone Number
ext	Physical Contact Information	*unstructured*	Telephone Extension
comment	Physical Contact Information	*unstructured*	Telephone Optional Comments

5.5.6 Contact Information

The **contact** structure is used to specify contact information. Services can specify precisely which set of data they need, postal, telecommunication, or online address information.

`contact`	**Category**	**Structure**	**Short display name**
postal	Physical Contact Information, Demographic and Socioeconomic Data	postal	Postal Address Information
telecom	Physical Contact Information	telecom	Telecommunications Information
online	Online Contact Information	online	Online Address Information

5.5.6.1 Postal

The **postal** structure specifies a postal mailing address.

`postal`	**Category**	**Structure**	**Short display name**
name	Physical Contact Information, Demographic and Socioeconomic Data	personname	Name
street	Physical Contact Information	*unstructured*	Street Address
city	Demographic and Socioeconomic Data	*unstructured*	City
stateprov	Demographic and Socioeconomic Data	*unstructured*	State or Province
postalcode	Demographic and Socioeconomic Data	*unstructured*	Postal Code
country	Demographic and Socioeconomic Data	*unstructured*	Country Name
organization	Demographic and Socioeconomic Data	*unstructured*	Organization Name

The "country" field represents the information of the name of the country (for example, one among the countries listed in [ISO3166]).

5.5.6.2 Telecommunication

The **telecom** structure specifies telecommunication information about a person.

`telecom`	**Category**	**Structure**	**Short display name**
telephone	Physical Contact Information	telephonenum	Telephone Number
fax	Physical Contact Information	telephonenum	Fax Number
mobile	Physical Contact Information	telephonenum	Mobile Telephone Number
pager	Physical Contact Information	telephonenum	Pager Number

5.5.6.3 Online

The **online** structure specifies online information about a person or legal entity.

`online`	**Category**	**Structure**	**Short display name**
email	Online Contact Information	*unstructured*	Email Address
uri	Online Contact Information	*unstructured*	Home Page Address

5.5.7 Access Logs and Internet Addresses

Two structures used for representing forms of Internet addresses are provided. The `uri` structure covers Universal Resource Identifiers (URI), which are defined in [URI]. The `ipaddr` structure represents IP addresses and Domain Name System (DNS) hostnames.

5.5.7.1 URI

`uri`	**Category**	**Structure**	**Short display name**
authority	*(variable-category)*	*unstructured*	URI Authority
stem	*(variable-category)*	*unstructured*	URI Stem
querystring	*(variable-category)*	*unstructured*	Query-string Portion of URI

The authority of a URI is defined as the `authority` component in [URI]. The stem of a URI is defined as the information contained in the portion of the URI after the authority and up to (and including) the first '?' character in the URI, and the querystring is the information contained in the portion of the URI after the first '?' character. For URIs which do not contain a '?' character, the stem is the entire URI, and the querystring is empty.

Since URI information can be used in different ways, depending on the context, all the fields in the `uri` structure are tagged as being of "variable" category. Schema definitions MUST explicitly set the corresponding category in the element referencing this data structure.

5.5.7.2 ipaddr

The `ipaddr` structure represents the hostname and IP address of a system.

`ipaddr`	**Category**	**Structure**	**Short display name**
hostname	Computer Information	*unstructured*	Complete Host and Domain Name
partialhostname	Demographic	*unstructured*	Partial Hostname
fullip	Computer Information	*unstructured*	Full IP Address
partialip	Demographic	*unstructured*	Partial IP Address

The `hostname` element is used to represent collection of either the simple hostname of a system, or the full hostname including domain name. The `partialhostname` element represents the information of a fully-qualified hostname which has had *at least* the host portion removed from the hostname. In other words, everything up to the first '.' in the fully-qualified hostname MUST be removed for an address to quality as a "partial hostname".

The `fullip` element represents the information of a full IP version 4 or IP version 6 address. The `partialip` element represents an IP version 4 address (only - not a version 6 address) which has had *at least* the last 7 bits of information removed. This removal MUST be done by replacing those bits with a fixed pattern for all visitors (for example, all 0's or all 1's).

Certain Web sites are known to make use not of the visitor's entire IP address or hostname, but rather make use of a reduced form of that information. By collecting only a subset of the address information, the site visitor is given some measure of anonymity. It is certainly not the intent of this specification to claim that these "stripped" IP addresses or hostnames are impossible to associate with an individual user, but rather that it is significantly more difficult to do so. Sites which perform this data reduction MAY wish to declare this practice in order to more-accurately reflect their practices.

5.5.7.3 Access Log Information

The `loginfo` structure is used to represent information typically stored in Web-server access logs.

`loginfo`	**Category**	**Structure**	**Short display name**
uri	Navigation and click-stream data	uri	URI of Requested Resource
timestamp	Navigation and click-stream data	date	Request Timestamp
clientip	Computer Information, Demographic and Socioeconomic Data	ipaddr	Client's IP Address or Hostname
other.httpmethod	Navigation and click-stream data	*unstructured*	HTTP Request Method
other.bytes	Navigation and click-stream data	*unstructured*	Data Bytes in Response
other.statuscode	Navigation and click-stream data	*unstructured*	Response Status Code

The resource in the HTTP request is captured by the `uri` field. The time at which the server processes the request is represented by the `timestamp` field. Server implementations are free to define this field as the time the request was received, the time that the server began sending the response, the time that sending the response was complete, or some other convenient representation of the time the request was processed. The IP address of the client system making the request is given by the `clientip` field.

The `other` data fields represent other information commonly stored in Web server access logs. `other.httpmethod` is the HTTP method (such as `GET`, `POST`, etc) in the client's request. `other.bytes` indicates the number of bytes in the response-body sent by the server. `other.statuscode` is the HTTP status code on the request, such as 200, 302, or 404 (see section 6.1.1 of [HTTP1.1] for details).

5.5.7.4 Other HTTP Protocol Information

The `httpinfo` structure represents information carried by the HTTP protocol which is not covered by the `loginfo` structure.

`httpinfo`	**Category**	**Structure**	**Short display name**
referer	Navigation and click-stream data	uri	Last URI Requested by the User
useragent	Computer Information	*unstructured*	User Agent Information

The `useragent` field represents the information in the HTTP `User-Agent` header (which gives information about the type and version of the user's Web browser), and/or the HTTP `accept`* headers.

The `referer` field represents the information in the HTTP `Referer` header, which gives information about the previous page visited by the user. Note that this field is misspelled in exactly the same way as the corresponding HTTP header.

5.6 The base data schema

All P3P-compliant user agent implementations MUST be aware of the data elements in the P3P base data schema. The P3P base data schema includes the definition of the basic data structures, and four data element sets: **`user`**, **`thirdparty`**, **`business`** and **`dynamic`**. The `user`, `thirdparty` and `business` sets include elements that users and/or businesses might provide values for, while the `dynamic` set includes elements that are dynamically generated in the course of a user's browsing session. User agents may support a variety of mechanisms that allow users to provide values for the elements in the `user` set and store them in a data repository, including mechanisms that support multiple personae. Users may choose not to provide values for these data elements.

The formal XML definition of the P3P base data schema is given in Appendix 3. In the following sections, the base data elements and sets are explained one by one. In the future there will be in all likelihood *demand for the creation of other data sets and elements.* Obvious applications include catalogue, payment, and agent/system attribute schemas (an extensive set of system elements is provided for example in http://www.w3.org/TR/NOTE-agent-attributes.)

Each table below specifies a **set**, the elements within the set, the category associated with the element, its structure, and the display name shown to users. More than one category may be associated with a fixed data element. However, each base data element is assigned to only one category whenever possible. It is recommended that data schema designers do the same.

5.6.1 User Data

The **`user`** data set includes general information about the user.

`user`	Category	Structure	Short display name
name	Physical Contact Information, Demographic and Socioeconomic Data	personname	User's Name
bdate	Demographic and Socioeconomic Data	date	User's Birth Date
login	Unique Identifiers	login	User's Login Information
cert	Unique Identifiers	certificate	User's Identity Certificate
gender	Demographic and Socioeconomic Data	*unstructured*	User's Gender (Male or Female)
employer	Demographic and Socioeconomic Data	*unstructured*	User's Employer
department	Demographic and Socioeconomic Data	*unstructured*	Department or Division of Organization where User is Employed
jobtitle	Demographic and Socioeconomic Data	*unstructured*	User's Job Title
home-info	Physical Contact Information, Online Contact Information, Demographic and Socioeconomic Data	contact	User's Home Contact Information
business-info	Physical Contact Information, Online Contact Information, Demographic and Socioeconomic Data	contact	User's Business Contact Information

Note, that this data set includes elements that are actually sets of data themselves. These sets are defined in the Data Structures subsection of this document. The short display name for an individual element contained within a data set is defined as the concatenation of the short display names that have been defined for the set and the element, separated by a separator appropriate for the language/script in question, e.g. a comma for English. For example, the short display name for `user.home-info.postal.postalcode` could be "User's Home Contact Information, Postal Address Information, Postal code". User agent implementations may prefer to develop their own short display names rather than using the concatenated names when displaying information for the user.

5.6.2 Third Party Data

The **`thirdparty`** data set allows users and businesses to provide values for a related third party. This can be useful whenever third party information needs to be exchanged, for example when ordering a present online that should be sent to another person, or when providing information about one's spouse or business partner. Such information could be stored in a user repository alongside the `user` data set. User agents may offer to store multiple such `thirdparty` data sets and allow users to select the appropriate values from a list when necessary.

The `thirdparty` data set is identical with the `user` data set. See section 5.6.1 User Data for details.

5.6.3 Business Data

The **`business`** data set features a subset of `user` data relevant for describing legal entities. In P3P1.0, this data set is primarily used for declaring the policy entity, although it should also be applicable to business-to-business interactions.

`business`	Category	Structure	Short display name
name	Demographic and Socioeconomic Data	*unstructured*	Organization Name
department	Demographic and Socioeconomic Data	*unstructured*	Department or Division of Organization
cert	Unique Identifiers	certificate	Organization Identity Certificate
contact-info	Physical Contact Information, Online Contact Information, Demographic and Socioeconomic Data	contact	Contact Information for the Organization

5.6.4 Dynamic Data

In some cases, there is a need to specify data elements that do not have fixed values that a user might type in or store in a repository. In the P3P base data schema, all such elements are grouped under the `dynamic` data set. Sites may refer to the types of data they collect using the dynamic data set only, rather than enumerating all of the specific data elements.

`dynamic`	Category	Structure	Short display name
clickstream	Navigation and Click-stream Data, Computer Information	loginfo	Click-stream Information
http	Navigation and Click-stream Data, Computer Information	httpinfo	HTTP Protocol Information
clientevents	Navigation and Click-stream Data	*unstructured*	User's Interaction with a Resource
cookies	*(variable-category)*	*unstructured*	Use of HTTP Cookies
miscdata	*(variable-category)*	*unstructured*	Miscellaneous Non-base Data Schema Information
searchtext	Interactive Data	*unstructured*	Search Terms
interactionrecord	Interactive Data	*unstructured*	Server Stores the Transaction History

These elements are often implicit in navigation or Web interactions. They should be used with categories to describe the type of information collected through these methods. A brief description of each element follows.

`clickstream`

The `clickstream` element is expected to apply to practically all Web sites. It represents the combination of information typically found in Web server access logs: the IP address or hostname of the user's computer, the URI of the resource requested, the time the request was made, the HTTP method used in the request, the size of the response, and the HTTP status code in the response. Web sites that collect standard server access logs as well as sites which do URI path analysis can use this data element to describe how that data will be used. Web sites that collect only some of the data elements listed for the `clickstream` element MAY choose to list those specific elements rather than the entire `dynamic.clickstream` element. This allows sites with more limited data-collection practices to accurately present those practices to their visitors.

`http`

The `http` element contains additional information contained in the HTTP protocol. See the definition of the `httpinfo` structure for descriptions of specific elements. Sites MAY use the `dynamic.http` field as a shorthand to cover all the elements in the `httpinfo` structure if they wish, or they MAY reference the specific elements in the `httpinfo` structure.

`clientevents`

The `clientevents` element represents data about how the user interacts with their Web browser while interacting with a resource. For example, an application may wish to collect information about whether the user moved their mouse over a certain image on a page, or whether the user ever brought up the help window in a Java applet. This kind of information is represented by the dynamic.clientevents data element. Much of this interaction record is represented by the events and data defined by the Document Object Model (DOM) Level 2 Events [DOM2-Events]. The `clientevents` data element also covers any other data regarding the user's interaction with their browser while the browser is displaying a resource. The exception is events which are covered by other elements in the base data schema. For example, requesting a page by clicking on a link is part of the user's interaction with their browser while viewing a page, but merely collecting the URL the user has clicked on does not require declaring this data element; `clickstream` covers that event. However, the DOM event `DOMFocusIn` (representing the user moving their mouse over an object on a page) is not covered by any other existing element, so if a site is collecting the occurrence of this event, then it needs to state that it collects the dynamic.clientevents element. Items covered by this data element are typically collected by client-side scripting languages, such as JavaScript, or by client-side applets, such as ActiveX or Java applets. Note that while the previous discussion has been in terms of a user viewing a resource, this data element also applies to Web applications which do not display resources visually - for example, audio-based Web browsers.

`cookies`

The `cookies` element should be used whenever HTTP cookies are set or retrieved by a site. Please note that `cookies` is a *variable data element* and requires the explicit declaration of usage categories in a policy.

`miscdata`

The `miscdata` element references information collected by the service that the service does not reference using a specific data element. Categories have to be used to better describe these data: sites MUST reference a separate `miscdata` element in their policies for each category of miscellaneous data they collect.

`searchtext`

The `searchtext` element references a specific type of solicitation used for searching and indexing sites. For example, if the only fields on a search engine page are search fields, the site only needs to disclose that data element.

`interactionrecord`

The `interactionrecord` element should be used if the server is keeping track of the interaction it has with the user (i.e. information other than clickstream data, for example account transactions, etc).

5.7 Categories and Data Elements/Structures

5.7.1 Fixed-Category Data Elements/Structures

Most of the elements in the base data schema are so called *"fixed"* data elements: they belong to one or at most two category classes. By assigning a category invariably to elements or structures in the base data schema, services and users are able to refer to entire groups of elements simply by referencing the corresponding category. For example, using [APPEL], the privacy preferences exchange language, users can write rules that warn them when they visit a site that collects any data element in a certain category.

When creating data schemas for fixed data elements, schema creators have to explicitly enumerate the categories that these element belong to. For example:

```
<DATA-STRUCT name="postal.street"     structref="#text"
          short-description="Street Address">
<CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>
```

If an element or structure belongs to multiple categories, multiple elements referencing the appropriate categories can be used. For example, the following piece of XML can be used to declare that the data elements in user.name have both category "physical" and "demographic":

```
<DATA-STRUCT name="user.name"     structref="#personname"
          short-description="User's Name">
<CATEGORIES><physical/><demographic/></CATEGORIES>
</DATA-STRUCT>
```

Please note that the category classes of fixed data elements/structures can **not** be overridden, for example by writing rules or policies that assign a different category to a known fixed base data element. User agents MUST ignore such categories and instead use the original category (or set of categories) listed in the schema definition. User agents MAY preferably alert the user that a fixed data element is used together with a non-standard category class.

5.7.2 Variable-Category Data Elements/Structures

Not all data elements/structures in the base data schema belong to a pre-determined category class. Some can contain information from a range of categories, depending on a particular situation. Such elements/structures are called *variable-category data elements/structures* (or "variable data element/structure" for short). Although most variable data elements in the P3P base data schema are combined in the **dynamic** element set, they can appear in any data set, even mixed with *fixed-category data elements*.

When creating a schema definition for such elements and/or structures, schema authors MUST NOT list an explicit category attribute, otherwise the element/structure becomes *fixed*. For example when specifying the "Year" *Data Structure*, which can take various categories depending on the situation (e.g. when used for a credit card expiration date vs. for a birth date), the following schema definition can be used:

```
<DATA-STRUCT name="date.ymd.year"
          short-description="Year"/>  <!-- Variable Data Structure-->
```

This allows new schema extensions that reference such variable-category *Data Structures* to assign a specific category to derived elements, depending on their usage in that extension. For example, an e-commerce schema extension could thus define a credit card expiration date as follows:

```
<DATA-STRUCT name="Card.ExpDate"         structref="#date.ymd"
          short-description="Card Expiration Date">
<CATEGORIES><purchase/></CATEGORIES>
</DATA-STRUCT>
```

Under these conditions, the variable Data Structure **date** is assigned a fixed category "Purchase Information" when being used for specifying a credit card expiration date.

Note that while user preferences can list such variable data elements without any additional category information (effectively expressing preferences over *any* usage of this element), services MUST always explicitly specify the categories that apply to the usage of a variable data element in their particular policy. This information has to appear as a category element in the corresponding `DATA` element listed in the policy, for example as in:

```
<POLICY ... >
   ...
   <DATA ref="#dynamic.cookies"><CATEGORIES><uniqueid/></CATEGORIES></DATA>
   ...
</POLICY>
```

where a service declares that cookies are used to recognize the user at this site (i.e. category Unique Identifiers).

If a service wants to declare a data element that is in multiple categories, it simply declares the corresponding categories (as shown in the above section):

```
<POLICY ... >
   ...
   <DATA ref="#dynamic.cookies"><CATEGORIES><uniqueid/><preference/></CATEGORIES></DATA>
   ...
</POLICY>
```

With the above declaration a service announces that it uses cookies both to recognize the user at this site *and* for storing user preference data. Note that for the purpose of P3P there is no difference whether this information is stored in two separate cookies or in a single one.

Finally, note that categories can be inherited as well: *Categories inherit downward when a field is structured, but only into fields which have no predefined category.* Therefore, we suggest to schema authors that they do their best to insure that all applicable categories are applied to new data elements they create.

5.6 Using Data Elements

P3P offers Web sites a great deal of flexibility in how they describe the types of data they collect.

- Sites may describe data generally using the **dynamic.miscdata** element and the appropriate categories.
- Sites may describe data specifically using the data elements defined in the base data schema.
- Sites may describe data specifically using data elements defined in new data schemas.

Any of these three methods may be combined within a single policy.

By using the **`dynamic.miscdata`** element, sites can specify the types of data they collect without having to enumerate every individual data element. This may be convenient for sites that collect a lot of data or sites belonging to large organizations that want to offer a single P3P policy covering the entire organization. However, the disadvantage of this approach is that user agents will have to assume that the site might collect any data element belonging to the categories referenced by the site. So, for example, if a site's policy states that it collects **`dynamic.miscdata`** of the physical contact information category, but the only physical contact information it collects is business address, user agents will nonetheless assume that the site might also collect telephone numbers. If the site wishes to be clear that it does not collect telephone numbers or any other physical contact information other than business address, than it should disclose that it collects **`user.business-info.contact.postal`**. Furthermore, as user agents are developed with automatic form-filling capabilities, it is likely that sites that enumerate the data they collect will be able to better integrate with these tools.

By defining new data schemas, sites can precisely specify the data they collect beyond the base data set. However, if user agents are unfamiliar with the elements defined in these schemas, they will be able to provide only minimal information to the user about these new elements. The information they provide will be based on the category and display names specified for each element.

Regardless of whether a site wishes to make general or specific data disclosures, there are additional advantages to disclosing specific elements from the **`dynamic`** data set. For example, by disclosing **`dynamic.cookies`** a site can indicate that it uses cookies and explain the purpose of this use. User agent implementations that offer users cookie control interfaces based on this information are encouraged. Likewise, user agents that by default do not send the HTTP_REFERER header, might look for the **`dynamic.http.referer`** element in P3P policies and send the header if it will be used for a purpose the user finds acceptable.

6. Appendices

Appendix 1: References (Normative)

[ABNF]
D. Crocker, P. Overel. "Augmented BNF for Syntax Specifications: ABNF," RFC2234, IETF, November 1997.
Available at http://www.ietf.org/rfc/rfc2234.txt.

[CHARMODEL]
M. Dürst, *et al.* (Eds.), "Character Model for the World Wide Web," World Wide Web Consortium Working Draft. 20 February 2002.
Latest version available at http://www.w3.org/TR/charmod/.

[DOM2-Events]
T. Pixley (Ed.), "Document Object Model (DOM) Level 2 Events Specification," World Wide Web Consortium, Recommendation. 13 November 2000.
Available at http://www.w3.org/TR/DOM-Level-2-Events/.

[HTTP1.0]

T. Berners-Lee, R. Fielding, H. Frystyk, "Hypertext Transfer Protocol -- HTTP/1.0," RFC1945, IETF, May 1996.
Available at http://www.ietf.org/rfc/rfc1945.txt.

[HTTP1.1]
R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," RFC2616, IETF, June 1999. [Updates RFC2068]
Available at http://www.ietf.org/rfc/rfc2616.txt.

[HTML]
D. Raggett, A. Le Hors, and I. Jacobs (Eds.). "HTML 4.01 Specification" World Wide Web Consortium, Recommendation. 24 Dicember 1999.
Available at http://www.w3.org/TR/html401/.

[KEY]
S. Bradner. "Key words for use in RFCs to Indicate Requirement Levels." RFC2119, IETF, March 1997.
Available at http://www.ietf.org/rfc/rfc2119.txt.

[LANG]
H. Alvestrand, "Tags for the Identification of Languages." RFC1766, IETF, 1995.
Available at http://www.ietf.org/rfc/rfc1766.txt.

[STATE]
D. Kristol, L. Montulli, "HTTP State Management Mechanism." RFC2695, IETF, October, 2000 [Obsoletes RFC2109]
Available at http://www.ietf.org/rfc/rfc2965.txt.

[URI]
T. Berners-Lee, R. Fielding, and L. Masinter. "Uniform Resource Identifiers (URI): Generic Syntax and Semantics." RFC2396, IETF, August 1998. [Updates RFC1738]
Available at http://www.ietf.org/rfc/rfc2396.txt.

[UTF-8]
F. Yergeau. "UTF-8, a transformation format of ISO 10646." RFC2279, IETF, January 1998.
Available at http://www.ietf.org/rfc/rfc2279.txt.

[XHTML-MOD]
M. Altheim, *et al.* (Eds.). "Modularization of XHTML". World Wide Web Consortium, Recommendation. 10 April 2000.
Available at http://www.w3.org/TR/xhtml-modularization/.

[XML]
T. Bray, J. Paoli, C. M. Sperberg-McQueen, E.Maler (Eds.). "Extensible Markup Language (XML) 1.0 Specification (Second Edition)." World Wide Web Consortium, Recommendation. 6 October 2000.
Available at http://www.w3.org/TR/REC-xml.

[XML-Name]
T. Bray, D. Hollander, A. Layman (Eds.). "Namespaces in XML." World Wide Web Consortium, Recommendation. 14 January 1999.
Available at http://www.w3.org/TR/REC-xml-names/.

[XML-Schema1]
H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn (Eds.). "XML Schema Part 1: Structures" World Wide Web Consortium Recommendation. 2 May 2001.
Available at http://www.w3.org/TR/xmlschema-1/.

[XML-Schema2]
P. Biron, A. Malhotra (Eds.). "XML Schema Part 2: Datatypes" World Wide Web Consortium Recommendation. 2 May 2001.
Available at http://www.w3.org/TR/xmlschema-2/.

Appendix 2: References (Non-Normative)

[APPEL]
M. Langheinrich (Ed.). "A P3P Preference Exchange Language (APPEL)." World Wide Web Consortium Working Draft. 26 February 2001.
Available at http://www.w3.org/TR/P3P-preferences.

[CACHING]
I. Cooper, I. Melve, G. Tomlinson. "Internet Web Replication and Caching Taxonomy." RFC3040, IETF, January 2001.
Available at http://www.ietf.org/rfc/rfc3040.txt.

[COOKIES]
"Persistent Client State -- HTTP Cookies," Preliminary Specification, Netscape, 1999.
Available at http://www.netscape.com/newsref/std/cookie_spec.html.

[ISO3166]
"ISO3166: Codes for The Representation of Names of Countries." International Organization for Standardization.

[ISO8601]
"ISO8601: Data elements and interchange formats -- Information interchange -- Representation of dates and times." International Organization for Standardization.

[P3P-HEADER]
M. Marchiori, R. Lotenberg (Eds.), "The HTTP header for the Platform for Privacy Preferences 1.0 (P3P1.0)." IETF Internet Draft, 2002.
Latest version available as text at http://www.w3.org/2002/04/P3Pv1-header.txt.
Latest version available as HTML at http://www.w3.org/2002/04/P3Pv1-header.html.
Latest version available as XML at http://www.w3.org/2002/04/P3Pv1-header.xml.

[P3P-RDF]
B. McBride, R.Wenning, L.Cranor. "An RDF Schema for P3P." World Wide Web Consortium, Note. 25 January 2002.
Latest version available at http://www.w3.org/TR/p3p-rdfschema/.

[RDF]
O. Lassila and R. Swick (Eds.). "Resource Description Framework (RDF) Model and Syntax Specification." World Wide Web Consortium, Recommendation. 22 February 1999.
Available at http://www.w3.org/TR/REC-rdf-syntax/.

[UNICODE]
Unicode Consortium. "The Unicode Standard"
Available at http://www.unicode.org/unicode/standard/standard.html.

Appendix 3: The P3P base data schema Definition (Normative)

The data schema corresponding to the P3P base data schema follows for easy reference. The schema is also present as a separate file at the URI http://www.w3.org/TR/P3P/base .

```
<DATASCHEMA xmlns="http://www.w3.org/2002/01/P3Pv1">
<!-- ********** Base Data Structures ********** -->

<!-- "date" Data Structure -->
```

```
<DATA-STRUCT name="date.ymd.year"
    short-description="Year"/>

<DATA-STRUCT name="date.ymd.month"
    short-description="Month"/>

<DATA-STRUCT name="date.ymd.day"
    short-description="Day"/>

<DATA-STRUCT name="date.hms.hour"
    short-description="Hour"/>

<DATA-STRUCT name="date.hms.minute"
    short-description="Minutes"/>

<DATA-STRUCT name="date.hms.second"
    short-description="Second"/>

<DATA-STRUCT name="date.fractionsecond"
    short-description="Fraction of Second"/>

<DATA-STRUCT name="date.timezone"
    short-description="Time Zone"/>

<!-- "login" Data Structure -->
<DATA-STRUCT name="login.id"
    short-description="Login ID">
    <CATEGORIES><uniqueid/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="login.password"
    short-description="Login Password">
    <CATEGORIES><uniqueid/></CATEGORIES>
</DATA-STRUCT>

<!-- "personname" Data Structure -->
<DATA-STRUCT name="personname.prefix"
    short-description="Name Prefix">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="personname.given"
    short-description="Given Name (First Name)">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="personname.middle"
    short-description="Middle Name">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="personname.family"
    short-description="Family Name (Last Name)">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="personname.suffix"
    short-description="Name Suffix">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="personname.nickname"
    short-description="Nickname">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<!-- "certificate" Data Structure -->
<DATA-STRUCT name="certificate.key"
    short-description="Certificate key">
    <CATEGORIES><uniqueid/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="certificate.format"
    short-description="Certificate format">
    <CATEGORIES><uniqueid/></CATEGORIES>
</DATA-STRUCT>

<!-- "telephonenum" Data Structure -->
<DATA-STRUCT name="telephonenum.intcode"
    short-description="International Telephone Code">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="telephonenum.loccode"
    short-description="Local Telephone Area Code">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="telephonenum.number"
    short-description="Telephone Number">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="telephonenum.ext"
    short-description="Telephone Extension">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="telephonenum.comment"
    short-description="Telephone Optional Comments">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<!-- "postal" Data Structure -->
<DATA-STRUCT name="postal.name" structref="#personname">
</DATA-STRUCT>

<DATA-STRUCT name="postal.street"
    short-description="Street Address">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="postal.city"
    short-description="City">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="postal.stateprov"
    short-description="State or Province">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="postal.postalcode"
    short-description="Postal Code">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="postal.organization"
    short-description="Organization Name">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="postal.country"
    short-description="Country Name">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<!-- "telecom" Data Structure -->
<DATA-STRUCT name="telecom.telephone"
    short-description="Telephone Number"
    structref="#telephonenum">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="telecom.fax"
    short-description="Fax Number"
    structref="#telephonenum">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="telecom.mobile"
    short-description="Mobile Telephone Number"
    structref="#telephonenum">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="telecom.pager"
    short-description="Pager Number"
    structref="#telephonenum">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<!-- "online" Data Structure -->
<DATA-STRUCT name="online.email"
    short-description="Email Address">
    <CATEGORIES><online/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="online.uri"
    short-description="Home Page Address">
    <CATEGORIES><online/></CATEGORIES>
</DATA-STRUCT>

<!-- "contact" Data Structure -->
<DATA-STRUCT name="contact.postal"
    short-description="Postal Address Information"
    structref="#postal">
</DATA-STRUCT>

<DATA-STRUCT name="contact.telecom"
    short-description="Telecommunications Information"
    structref="#telecom">
    <CATEGORIES><physical/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="contact.online"
    short-description="Online Address Information"
    structref="#online">
    <CATEGORIES><online/></CATEGORIES>
</DATA-STRUCT>

<!-- "uri" Data Structure -->
<DATA-STRUCT name="uri.authority"
    short-description="URI Authority"/>

<DATA-STRUCT name="uri.stem"
    short-description="URI Stem"/>

<DATA-STRUCT name="uri.querystring"
    short-description="Query-string Portion of URI"/>
```

```
<!-- "ipaddr" Data Structure -->
<DATA-STRUCT name="ipaddr.hostname"
    short-description="Complete Host and Domain Name">
    <CATEGORIES><computer/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="ipaddr.partialhostname"
    short-description="Partial Hostname">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="ipaddr.fullip"
    short-description="Full IP Address">
    <CATEGORIES><computer/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="ipaddr.partialip"
    short-description="Partial IP Address">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-STRUCT>

<!-- "loginfo" Data Structure -->
<DATA-STRUCT name="loginfo.uri"
    short-description="URI of Requested Resource"
    structref="#uri">
    <CATEGORIES><navigation/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="loginfo.timestamp"
    short-description="Request Timestamp"
    structref="#date">
    <CATEGORIES><navigation/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="loginfo.clientip"
    short-description="Client's IP Address or Hostname"
    structref="#ipaddr">
</DATA-STRUCT>

<DATA-STRUCT name="loginfo.other.httpmethod"
    short-description="HTTP Request Method">
    <CATEGORIES><navigation/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="loginfo.other.bytes"
    short-description="Data Bytes in Response">
    <CATEGORIES><navigation/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="loginfo.other.statuscode"
    short-description="Response Status Code">
    <CATEGORIES><navigation/></CATEGORIES>
</DATA-STRUCT>

<!-- "httpinfo" Data Structure -->
<DATA-STRUCT name="httpinfo.referer"
    short-description="Last URI Requested by the User"
    structref="#uri">
    <CATEGORIES><navigation/></CATEGORIES>
</DATA-STRUCT>

<DATA-STRUCT name="httpinfo.useragent"
    short-description="User Agent Information">
    <CATEGORIES><computer/></CATEGORIES>
</DATA-STRUCT>

<!-- ********** Base Data Schemas ********** -->

<!-- "dynamic" Data Schema -->
<DATA-DEF name="dynamic.clickstream"
    short-description="Click-stream Information"
    structref="#loginfo">
    <CATEGORIES><navigation/><computer/><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="dynamic.http"
    short-description="HTTP Protocol Information"
    structref="#httpinfo">
    <CATEGORIES><navigation/><computer/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="dynamic.clientevents"
    short-description="User's Interaction with a Resource">
    <CATEGORIES><navigation/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="dynamic.cookies"
    short-description="Use of HTTP Cookies"/>

<DATA-DEF name="dynamic.searchtext"
    short-description="Search Terms">
    <CATEGORIES><interactive/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="dynamic.interactionrecord"
    short-description="Server Stores the Transaction History">
    <CATEGORIES><interactive/></CATEGORIES>
</DATA-DEF>
```

```
<DATA-DEF name="dynamic.miscdata"
    short-description="Miscellaneous Non-base Data Schema =
information"/>

<!-- "user" Data Schema -->
<DATA-DEF name="user.name"
    short-description="User's Name"
    structref="#personname">
    <CATEGORIES><physical/><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="user.bdate"
    short-description="User's Birth Date"
    structref="#date">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="user.login"
    short-description="User's Login Information"
    structref="#login">
    <CATEGORIES><uniqueid/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="user.cert"
    short-description="User's Identity Certificate"
    structref="#certificate">
    <CATEGORIES><uniqueid/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="user.gender"
    short-description="User's Gender">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="user.jobtitle"
    short-description="User's Job Title">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="user.home-info"
    short-description="User's Home Contact Information"
    structref="#contact">
    <CATEGORIES><physical/><online/><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="user.business-info"
    short-description="User's Business Contact Information"
    structref="#contact">
    <CATEGORIES><physical/><online/><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="user.employer"
    short-description="Name of User's Employer">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="user.department"
    short-description="Department or Division of Organization where User is Employed">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<!-- "thirdparty" Data Schema -->
<DATA-DEF name="thirdparty.name"
    short-description="Third Party's Name"
    structref="#personname">
    <CATEGORIES><physical/><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="thirdparty.bdate"
    short-description="Third Party's Birth Date"
    structref="#date">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="thirdparty.login"
    short-description="Third Party's Login Information"
    structref="#login">
    <CATEGORIES><uniqueid/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="thirdparty.cert"
    short-description="Third Party's Identity Certificate"
    structref="#certificate">
    <CATEGORIES><uniqueid/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="thirdparty.gender"
    short-description="Third Party's Gender">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="thirdparty.jobtitle"
    short-description="Third Party's Job Title">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="thirdparty.home-info"
    short-description="Third Party's Home Contact Information"
    structref="#contact">
    <CATEGORIES><physical/><online/><demographic/></CATEGORIES>
```

```
</DATA-DEF>

<DATA-DEF name="thirdparty.business-info"
    short-description="Third Party's Business Contact Information"
    structref="#contact">
    <CATEGORIES><physical/><online/><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="thirdparty.employer"
    short-description="Name of Third Party's Employer">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="thirdparty.department"
    short-description="Department or Division of Organization where Third Party is Employed">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<!-- "business" Data Schema -->
<DATA-DEF name="business.name"
    short-description="Organization Name">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="business.department"
    short-description="Department or Division of Organization">
    <CATEGORIES><demographic/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="business.cert"
    short-description="Organization Identity certificate"
    structref="#certificate">
    <CATEGORIES><uniqueid/></CATEGORIES>
</DATA-DEF>

<DATA-DEF name="business.contact-info"
    short-description="Contact Information for the Organization"
    structref="#contact">
    <CATEGORIES><physical/><online/><demographic/></CATEGORIES>
</DATA-DEF>

</DATASCHEMA>
```

Appendix 4: XML Schema Definition (Normative)

This appendix contains the XML schema for P3P policy reference files, for P3P policy documents, and for P3P data schema documents. P3P policy reference files, P3P policy documents and P3P data schema documents are XML documents that MUST conform to this schema. Note that this schema is based on the XML Schema specification [XML-Schema1][XML-Schema2]. The schema is also present as a separate file at the URI http://www.w3.org/2002/01/P3Pv1.xsd .

```
<?xml version='1.0' encoding='UTF-8'?>
<schema
  xmlns='http://www.w3.org/2001/XMLSchema'
  xmlns:p3p='http://www.w3.org/2002/01/P3Pv1'
  targetNamespace='http://www.w3.org/2002/01/P3Pv1'
  elementFormDefault='qualified'>

<!-- enabling xml:lang attribute -->
 <import namespace='http://www.w3.org/XML/1998/namespace'
    schemaLocation='http://www.w3.org/2001/xml.xsd' />

<!-- Basic P3P Data Type -->
 <simpleType name='yes_no'>
  <restriction base='string'>
   <enumeration value='yes'/>
   <enumeration value='no'/>
  </restriction>
 </simpleType>


<!-- *********** Policy Reference *********** -->
<!-- ************** META ************** -->
 <element name='META'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:POLICY-REFERENCES'/>
    <element ref='p3p:POLICIES' minOccurs='0'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
   <attribute ref='xml:lang' use='optional'/>
  </complexType>
 </element>

<!-- ******* POLICY-REFERENCES ******** -->
 <element name='POLICY-REFERENCES'>
  <complexType>
   <sequence>
    <element ref='p3p:EXPIRY' minOccurs='0'/>
    <element ref='p3p:POLICY-REF' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:HINT' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
  </sequence>
  </complexType>
 </element>

 <element name='POLICY-REF'>
```

```
  <complexType>
   <sequence>
    <element name='INCLUDE'
             minOccurs='0' maxOccurs='unbounded' type='anyURI'/>
    <element name='EXCLUDE'
             minOccurs='0' maxOccurs='unbounded' type='anyURI'/>
    <element name='COOKIE-INCLUDE'
             minOccurs='0' maxOccurs='unbounded' type='p3p:cookie-element'/>
    <element name='COOKIE-EXCLUDE'
             minOccurs='0' maxOccurs='unbounded' type='p3p:cookie-element'/>
    <element name='METHOD'
             minOccurs='0' maxOccurs='unbounded' type='anyURI'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
   <attribute name='about' type='anyURI' use='required'/>
  </complexType>
 </element>

 <complexType name='cookie-element'>
  <attribute name='name' type='string' use='optional'/>
  <attribute name='value' type='string' use='optional'/>
  <attribute name='domain' type='string' use='optional'/>
  <attribute name='path' type='string' use='optional'/>
 </complexType>

<!-- ************* HINT ************* -->
 <element name='HINT'>
  <complexType>
   <attribute name='scope' type='string' use='required'/>
   <attribute name='path' type='string' use='required'/>
  </complexType>
 </element>

<!-- ************ POLICIES ************ -->
 <element name='POLICIES'>
  <complexType>
   <sequence>
    <element ref='p3p:EXPIRY' minOccurs='0'/>
    <element ref='p3p:DATASCHEMA' minOccurs='0'/>
    <element ref='p3p:POLICY' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
   <attribute ref='xml:lang' use='optional'/>
  </complexType>
 </element>


<!-- ************* EXPIRY ************* -->
 <element name='EXPIRY'>
  <complexType>
   <attribute name='max-age' type='nonNegativeInteger' use='optional'/>
   <attribute name='date' type='string' use='optional'/>
  </complexType>
 </element>

<!-- **************** Policy **************** -->
<!-- ************* POLICY ************* -->
 <element name='POLICY'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:TEST' minOccurs='0'/>
    <element ref='p3p:ENTITY'/>
    <element ref='p3p:ACCESS'/>
    <element ref='p3p:DISPUTES-GROUP' minOccurs='0'/>
    <element ref='p3p:STATEMENT' maxOccurs='unbounded'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
   <attribute name='discuri' type='anyURI' use='required'/>
   <attribute name='opturi' type='anyURI' use='optional'/>
   <attribute name='name' type='ID' use='required'/>
   <attribute ref='xml:lang' use='optional'/>
  </complexType>
 </element>

<!-- ************* TEST ************* -->
 <element name='TEST'>
  <complexType/>
 </element>

<!-- ************* ENTITY ************* -->
 <element name='ENTITY'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element name='DATA-GROUP'>
     <complexType>
      <sequence>
       <element name='DATA' type='p3p:data-in-entity' maxOccurs='unbounded'/>
      </sequence>
     </complexType>
    </element>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <complexType name='data-in-entity' mixed='true'>
  <attribute name='ref' type='anyURI' use='required'/>
 </complexType>
```

```
<!-- ************* ACCESS ************* -->
 <element name='ACCESS'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice>
     <element name='nonident' type='p3p:access-value'/>
     <element name='ident-contact' type='p3p:access-value'/>
     <element name='other-ident' type='p3p:access-value'/>
     <element name='contact-and-other' type='p3p:access-value'/>
     <element name='all' type='p3p:access-value'/>
     <element name='none' type='p3p:access-value'/>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <complexType name='access-value'/>

<!-- ************ DISPUTES ************ -->
 <element name='DISPUTES-GROUP'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:DISPUTES' maxOccurs='unbounded'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <element name='DISPUTES'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice minOccurs='0'>
     <sequence>
      <element ref='p3p:LONG-DESCRIPTION'/>
      <element ref='p3p:IMG' minOccurs='0'/>
      <element ref='p3p:REMEDIES' minOccurs='0'/>
      <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
     </sequence>
     <sequence>
      <element ref='p3p:IMG'/>
      <element ref='p3p:REMEDIES' minOccurs='0'/>
      <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
     </sequence>
     <sequence>
      <element ref='p3p:REMEDIES'/>
      <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
     </sequence>
    </choice>
   </sequence>
   <attribute name='resolution-type' use='required'>
    <simpleType>
     <restriction base='string'>
      <enumeration value='service'/>
      <enumeration value='independent'/>
      <enumeration value='court'/>
      <enumeration value='law'/>
     </restriction>
    </simpleType>
   </attribute>
   <attribute name='service' type='anyURI' use='required'/>
   <attribute name='verification' type='string' use='optional'/>
   <attribute name='short-description' type='string' use='optional'/>
  </complexType>
 </element>

<!-- ******** LONG-DESCRIPTION ******** -->
 <element name='LONG-DESCRIPTION'>
  <simpleType>
   <restriction base='string'/>
  </simpleType>
 </element>

<!-- ************** IMG *************** -->
 <element name='IMG'>
  <complexType>
   <attribute name='src' type='anyURI' use='required'/>
   <attribute name='width' type='nonNegativeInteger' use='optional'/>
   <attribute name='height' type='nonNegativeInteger' use='optional'/>
   <attribute name='alt' type='string' use='required'/>
  </complexType>
 </element>

<!-- ************ REMEDIES ************ -->
 <element name='REMEDIES'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice maxOccurs='unbounded'>
     <element name='correct' type='p3p:remedies-value'/>
     <element name='money' type='p3p:remedies-value'/>
     <element name='law' type='p3p:remedies-value'/>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>
 <complexType name='remedies-value'/>

<!-- *********** STATEMENT ************ -->
 <element name='STATEMENT'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element name='CONSEQUENCE' minOccurs='0' type='string'/>
    <choice>
     <sequence>
      <element ref='p3p:PURPOSE'/>
      <element ref='p3p:RECIPIENT'/>
      <element ref='p3p:RETENTION'/>
      <element name='DATA-GROUP' type='p3p:data-group-type' maxOccurs='unbounded'/>
     </sequence>
     <sequence>
      <element name='NON-IDENTIFIABLE'/>
      <element ref='p3p:PURPOSE' minOccurs='0'/>
      <element ref='p3p:RECIPIENT' minOccurs='0'/>
      <element ref='p3p:RETENTION' minOccurs='0'/>
      <element name='DATA-GROUP' type='p3p:data-group-type' minOccurs='0' maxOccurs='unbounded'/>
     </sequence>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

<!-- ************ PURPOSE ************* -->
 <element name='PURPOSE'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice maxOccurs='unbounded'>
     <element name='current' type='p3p:purpose-value'/>
     <element name='admin' type='p3p:purpose-value'/>
     <element name='develop' type='p3p:purpose-value'/>
     <element name='tailoring' type='p3p:purpose-value'/>
     <element name='pseudo-analysis' type='p3p:purpose-value'/>
     <element name='pseudo-decision' type='p3p:purpose-value'/>
     <element name='individual-analysis' type='p3p:purpose-value'/>
     <element name='individual-decision' type='p3p:purpose-value'/>
     <element name='contact' type='p3p:purpose-value'/>
     <element name='historical' type='p3p:purpose-value'/>
     <element name='telemarketing' type='p3p:purpose-value'/>
     <element name='other-purpose'>
      <complexType mixed='true'>
       <attribute name='required' use='optional' type='p3p:required-value'/>
      </complexType>
     </element>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <simpleType name='required-value'>
  <restriction base='string'>
   <enumeration value='always'/>
   <enumeration value='opt-in'/>
   <enumeration value='opt-out'/>
  </restriction>
 </simpleType>

 <complexType name='purpose-value'>
  <attribute name='required' use='optional' type='p3p:required-value' default='always' />
 </complexType>

<!-- *********** RECIPIENT ************ -->
 <element name='RECIPIENT'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice maxOccurs='unbounded'>
     <element name='ours'>
      <complexType>
       <sequence>
        <element ref='p3p:recipient-description' minOccurs='0' maxOccurs='unbounded'/>
       </sequence>
      </complexType>
     </element>
     <element name='same' type='p3p:recipient-value'/>
     <element name='other-recipient' type='p3p:recipient-value'/>
     <element name='delivery' type='p3p:recipient-value'/>
     <element name='public' type='p3p:recipient-value'/>
     <element name='unrelated' type='p3p:recipient-value'/>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <complexType name='recipient-value'>
  <sequence>
   <element ref='p3p:recipient-description' minOccurs='0' maxOccurs='unbounded'/>
  </sequence>
  <attribute name='required' use='optional' type='p3p:required-value'/>
 </complexType>
 <element name='recipient-description'>
  <complexType mixed='true'/>
 </element>

<!-- *********** RETENTION ************ -->
 <element name='RETENTION'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice>
     <element name='no-retention' type='p3p:retention-value'/>
     <element name='stated-purpose' type='p3p:retention-value'/>
     <element name='legal-requirement' type='p3p:retention-value'/>
     <element name='indefinitely' type='p3p:retention-value'/>
     <element name='business-practices' type='p3p:retention-value'/>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <complexType name='retention-value'/>

<!-- ************** DATA ************** -->
 <complexType name='data-group-type'>
  <sequence>
   <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   <element name='DATA' type='p3p:data-in-statement' maxOccurs='unbounded'/>
   <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
  </sequence>
  <attribute name='base' type='anyURI'
             use='optional' default='http://www.w3.org/TR/P3P/base'/>
 </complexType>

 <complexType name='data-in-statement' mixed='true'>
  <sequence minOccurs='0' maxOccurs='unbounded'>
   <element ref='p3p:CATEGORIES'/>
  </sequence>
  <attribute name='ref' type='anyURI' use='required'/>
  <attribute name='optional' use='optional' default='no' type='p3p:yes_no'/>
 </complexType>

<!-- ************** Data Schema ************* -->
<!-- *********** DATASCHEMA *********** -->
 <element name='DATASCHEMA'>
  <complexType>
   <choice minOccurs='0' maxOccurs='unbounded'>
    <element ref='p3p:DATA-DEF'/>
    <element ref='p3p:DATA-STRUCT'/>
    <element ref='p3p:EXTENSION'/>
   </choice>
   <attribute ref='xml:lang' use='optional'/>
  </complexType>
 </element>

 <element name='DATA-DEF' type='p3p:data-def'/>
 <element name='DATA-STRUCT' type='p3p:data-def'/>

 <complexType name='data-def'>
  <sequence>
   <element ref='p3p:CATEGORIES' minOccurs='0'/>
   <element ref='p3p:LONG-DESCRIPTION' minOccurs='0'/>
  </sequence>
  <attribute name='name' type='ID' use='required'/>
  <attribute name='structref' type='anyURI' use='optional'/>
  <attribute name='short-description' type='string' use='optional'/>
 </complexType>

<!-- *********** CATEGORIES *********** -->
 <element name='CATEGORIES'>
  <complexType>
   <choice maxOccurs='unbounded'>
    <element name='physical' type='p3p:categories-value'/>
    <element name='online' type='p3p:categories-value'/>
    <element name='uniqueid' type='p3p:categories-value'/>
    <element name='purchase' type='p3p:categories-value'/>
    <element name='financial' type='p3p:categories-value'/>
    <element name='computer' type='p3p:categories-value'/>
    <element name='navigation' type='p3p:categories-value'/>
    <element name='interactive' type='p3p:categories-value'/>
    <element name='demographic' type='p3p:categories-value'/>
    <element name='content' type='p3p:categories-value'/>
    <element name='state' type='p3p:categories-value'/>
    <element name='political' type='p3p:categories-value'/>
    <element name='health' type='p3p:categories-value'/>
    <element name='preference' type='p3p:categories-value'/>
    <element name='location' type='p3p:categories-value'/>
    <element name='government' type='p3p:categories-value'/>
    <element name='other-category' type='string'/>
   </choice>
  </complexType>
 </element>

 <complexType name='categories-value'/>

<!-- *********** EXTENSION ************ -->
 <element name='EXTENSION'>
  <complexType mixed='true'>
   <choice minOccurs='0' maxOccurs='unbounded'>
    <any minOccurs='0' maxOccurs='unbounded' processContents='skip'/>
```

```
      </choice>
      <attribute name='optional' use='optional' default='yes' type='p3p:yes_no'/>
     </complexType>
    </element>

   </schema>
```

Appendix 5: XML DTD Definition (Non-normative)

This appendix contains the DTD for P3P policy reference files, for P3P policy documents, and for P3P data schema documents. This DTD MAY be used to verify that P3P files are valid (although, note that there are some valid files that may be rejected if checked against the DTD). The DTD is also present as a separate file at the URI http://www.w3.org/2002/01/P3Pv1.dtd .

```
<!-- *************** Entities *************** -->
<!ENTITY % URI "CDATA">
<!ENTITY % NUMBER "CDATA">

<!-- *********** Policy Reference *********** -->

<!-- ************** META ************** -->
<!ELEMENT META (EXTENSION*, POLICY-REFERENCES, POLICIES?, EXTENSION*)>
<!ATTLIST META xml:lang NMTOKEN #IMPLIED>
<!ATTLIST META xmlns CDATA #FIXED "http://www.w3.org/2002/01/P3Pv1">

<!-- ******* POLICY-REFERENCES ******** -->
<!ELEMENT POLICY-REFERENCES (EXPIRY?, POLICY-REF*, HINT*, EXTENSION*)>

<!-- *********** POLICY-REF *********** -->
<!ELEMENT POLICY-REF (INCLUDE*,
    EXCLUDE*,

    COOKIE-INCLUDE*,

    COOKIE-EXCLUDE*,
    METHOD*,
    EXTENSION*)>
<!ATTLIST POLICY-REF
    about %URI; #REQUIRED >

<!-- ************** HINT ************** -->
<!ELEMENT HINT EMPTY>
<!ATTLIST HINT
    scope  CDATA  #IMPLIED
    path   CDATA  #IMPLIED >

<!-- ************* EXPIRY ************* -->
<!ELEMENT EXPIRY EMPTY>
<!ATTLIST EXPIRY
    max-age %NUMBER; #IMPLIED
    date    CDATA    #IMPLIED >

<!-- ************ POLICIES ************ -->
<!ELEMENT POLICIES (EXPIRY?, DATASCHEMA?,
    POLICY*)>
<!ATTLIST POLICIES xml:lang NMTOKEN #IMPLIED>
<!ATTLIST POLICIES xmlns CDATA #FIXED "http://www.w3.org/2002/01/P3Pv1">

<!-- ***** INCLUDE/EXCLUDE/METHOD ***** -->
<!ELEMENT INCLUDE          (#PCDATA)>
<!ELEMENT EXCLUDE          (#PCDATA)>
<!ELEMENT COOKIE-INCLUDE   EMPTY>
<!ATTLIST COOKIE-INCLUDE
    name   CDATA  #IMPLIED
    value  CDATA  #IMPLIED
    domain CDATA  #IMPLIED
    path   CDATA  #IMPLIED>
<!ELEMENT COOKIE-EXCLUDE   EMPTY>
<!ATTLIST COOKIE-EXCLUDE
    name   CDATA  #IMPLIED
    value  CDATA  #IMPLIED
    domain CDATA  #IMPLIED
    path   CDATA  #IMPLIED>
<!ELEMENT METHOD           (#PCDATA)>

<!-- **************** Policy **************** -->

<!-- ************* POLICY ************* -->
<!ELEMENT POLICY (EXTENSION*,
    TEST?,
    ENTITY,
    ACCESS,
    DISPUTES-GROUP?,
    STATEMENT+,
    EXTENSION*)>
<!ATTLIST POLICY
    name     ID      #REQUIRED
    discuri  %URI;   #REQUIRED
    opturi   %URI;   #IMPLIED
    xml:lang NMTOKEN #IMPLIED>

<!-- ******** TEST ******** -->
<!ELEMENT TEST EMPTY>

<!-- ************* ENTITY ************* -->
<!ELEMENT ENTITY (EXTENSION*, DATA-GROUP, EXTENSION*)>

<!-- ************* ACCESS ************* -->
<!ELEMENT ACCESS (EXTENSION*,
   (nonident
    | all
    | contact-and-other
    | ident-contact
    | other-ident
    | none),
    EXTENSION*)>
<!ELEMENT nonident          EMPTY>
<!ELEMENT all               EMPTY>
<!ELEMENT contact-and-other EMPTY>
<!ELEMENT ident-contact     EMPTY>
<!ELEMENT other-ident       EMPTY>
<!ELEMENT none              EMPTY>

<!-- ************ DISPUTES ************ -->
<!ELEMENT DISPUTES-GROUP (EXTENSION*, DISPUTES+, EXTENSION*)>
<!ELEMENT DISPUTES (EXTENSION*,
    ( (LONG-DESCRIPTION, IMG?, REMEDIES?, EXTENSION*)
      | (IMG, REMEDIES?, EXTENSION*)
      | (REMEDIES, EXTENSION*) )?)>
<!ATTLIST DISPUTES
    resolution-type   (service | independent | court | law) #REQUIRED
    service           %URI;                                 #REQUIRED
    verification      CDATA                                 #IMPLIED
    short-description CDATA                                 #IMPLIED >

<!-- ******** LONG-DESCRIPTION ******** -->
<!ELEMENT LONG-DESCRIPTION (#PCDATA)>

<!-- ************** IMG *************** -->
<!ELEMENT IMG EMPTY>
<!ATTLIST IMG
    src    %URI;    #REQUIRED
    width  %NUMBER; #IMPLIED
    height %NUMBER; #IMPLIED
    alt    CDATA    #REQUIRED >

<!-- ************ REMEDIES ************ -->
<!ELEMENT REMEDIES (EXTENSION*, (correct | money | law)+, EXTENSION*)>
<!ELEMENT correct EMPTY>
<!ELEMENT money   EMPTY>
<!ELEMENT law     EMPTY>

<!-- *********** STATEMENT ************ -->
<!ELEMENT STATEMENT (EXTENSION*,
    CONSEQUENCE?,
    ((PURPOSE,RECIPIENT,RETENTION,DATA-GROUP+)|
     (NON-IDENTIFIABLE,PURPOSE?,RECIPIENT?,RETENTION?,DATA-GROUP*)),
    EXTENSION*)>

<!-- ********** CONSEQUENCE *********** -->
<!ELEMENT CONSEQUENCE (#PCDATA)>

<!-- ******** NON-IDENTIFIABLE ******** -->
<!ELEMENT NON-IDENTIFIABLE EMPTY>

<!-- ************ PURPOSE ************* -->
<!ELEMENT PURPOSE (EXTENSION*,
   (current
    | admin
    | develop
    | customization
    | tailoring
    | pseudo-analysis
    | pseudo-decision
    | individual-analysis
    | individual-decision
    | contact
    | historical
    | telemarketing
    | other-purpose)+,
    EXTENSION*)>

<!ENTITY % pur_att
         "required (always | opt-in | opt-out) #IMPLIED">
<!ELEMENT current             EMPTY>
<!ATTLIST current             %pur_att;>
<!ELEMENT admin               EMPTY>
<!ATTLIST admin               %pur_att;>
<!ELEMENT develop             EMPTY>
<!ATTLIST develop             %pur_att;>
<!ELEMENT customization       EMPTY>
<!ATTLIST customization       %pur_att;>
<!ELEMENT tailoring           EMPTY>
<!ATTLIST tailoring           %pur_att;>
<!ELEMENT pseudo-analysis     EMPTY>
<!ATTLIST pseudo-analysis     %pur_att;>
<!ELEMENT pseudo-decision     EMPTY>
<!ATTLIST pseudo-decision     %pur_att;>
<!ELEMENT individual-analysis EMPTY>
<!ATTLIST individual-analysis %pur_att;>
<!ELEMENT individual-decision EMPTY>
<!ATTLIST individual-decision %pur_att;>
<!ELEMENT contact             EMPTY>
<!ATTLIST contact             %pur_att;>
<!ELEMENT profiling           EMPTY>
<!ATTLIST profiling           %pur_att;>
<!ELEMENT historical          EMPTY>
<!ATTLIST historical          %pur_att;>
<!ELEMENT telemarketing       EMPTY>
```

```
<!ATTLIST telemarketing       %pur_att;>
<!ELEMENT other-purpose       (#PCDATA)>
<!ATTLIST other-purpose       %pur_att;>

<!-- *********** RECIPIENT ************ -->
<!ELEMENT RECIPIENT (EXTENSION*,
    (ours
    | same
    | other-recipient
    | delivery
    | public
    | unrelated)+,
    EXTENSION*)>
<!ELEMENT ours                  (recipient-description*)>
<!ELEMENT same                  (recipient-description*)>
<!ATTLIST same                  %pur_att;>
<!ELEMENT other-recipient       (recipient-description*)>
<!ATTLIST other-recipient       %pur_att;>
<!ELEMENT delivery              (recipient-description*)>
<!ATTLIST delivery              %pur_att;>
<!ELEMENT public                (recipient-description*)>
<!ATTLIST public                %pur_att;>
<!ELEMENT unrelated             (recipient-description*)>
<!ATTLIST unrelated             %pur_att;>
<!ELEMENT recipient-description (#PCDATA)>

<!-- *********** RETENTION ************ -->
<!ELEMENT RETENTION (EXTENSION*,
    (no-retention
    | stated-purpose
    | legal-requirement
    | indefinitely
    | business-practices),
    EXTENSION*)>
<!ELEMENT no-retention       EMPTY>
<!ELEMENT stated-purpose     EMPTY>
<!ELEMENT legal-requirement  EMPTY>
<!ELEMENT indefinitely       EMPTY>
<!ELEMENT business-practices EMPTY>

<!-- ************** DATA ************** -->
<!ELEMENT DATA-GROUP (EXTENSION*, DATA+, EXTENSION*)>
<!ATTLIST DATA-GROUP
    base     %URI;      "http://www.w3.org/TR/P3P/base" >
<!ELEMENT DATA (#PCDATA | CATEGORIES)*>
<!ATTLIST DATA
    ref      %URI;      #REQUIRED
    optional (yes | no) "no" >


<!-- *********** DATA SCHEMA *********** -->
<!ELEMENT DATASCHEMA (DATA-DEF | DATA-STRUCT | EXTENSION)*>
<!ATTLIST DATASCHEMA xml:lang NMTOKEN #IMPLIED>
<!ATTLIST DATASCHEMA xmlns CDATA #FIXED "http://www.w3.org/2002/01/P3Pv1">

<!ELEMENT DATA-DEF    (CATEGORIES?, LONG-DESCRIPTION?)>
<!ATTLIST DATA-DEF
    name              ID    #REQUIRED
    structref         %URI; #IMPLIED
    short-description CDATA #IMPLIED  >

<!ELEMENT DATA-STRUCT (CATEGORIES?, LONG-DESCRIPTION?)>
<!ATTLIST DATA-STRUCT
    name              ID    #REQUIRED
    structref         %URI; #IMPLIED
    short-description CDATA #IMPLIED  >

<!-- *********** CATEGORIES *********** -->
<!ELEMENT CATEGORIES (physical
  | online
  | uniqueid
  | purchase
  | financial
  | computer
  | navigation
  | interactive
  | demographic
  | content
  | state
  | political
  | health
  | preference
  | location
  | government
  | other-category)+>
<!ELEMENT physical    EMPTY>
<!ELEMENT online      EMPTY>
<!ELEMENT uniqueid    EMPTY>
<!ELEMENT purchase    EMPTY>
<!ELEMENT financial   EMPTY>
<!ELEMENT computer    EMPTY>
<!ELEMENT navigation  EMPTY>
<!ELEMENT interactive EMPTY>
<!ELEMENT demographic EMPTY>
<!ELEMENT content     EMPTY>
<!ELEMENT state       EMPTY>
<!ELEMENT political   EMPTY>
<!ELEMENT health      EMPTY>
<!ELEMENT preference  EMPTY>
<!ELEMENT location    EMPTY>
```

```
<!ELEMENT government  EMPTY>
<!ELEMENT other-category EMPTY>

<!-- *********** EXTENSION ************ -->
<!ELEMENT EXTENSION ANY>
<!ATTLIST EXTENSION
    optional (yes | no) "yes" >
```

Appendix 6: ABNF Notation (Normative)

The formal grammar of P3P is given in this specification using a slight modification of [ABNF]. The following is a simple description of the ABNF.

name = (elements)
where <name> is the name of the rule, <elements> is one or more rule names or terminals combined through the operands provided below. Rule names are case-insensitive.

(element1 element2)
elements enclosed in parentheses are treated as a single element, whose contents are strictly ordered.

<a>*<b>element
at least <a> and at most <b> occurrences of the element.
*(1*4<element> means one to four elements.)*

<a>element
exactly <a> occurrences of the element.
(4<element> means exactly 4 elements.)

<a>*element
<a> or more elements
(4<element> means 4 or more elements.)*

***<b>element**
0 to <b> elements.
*(*5<element> means 0 to 5 elements.)*

***element**
0 or more elements.
(<element> means 0 to infinite elements.)*

[element]
optional element, equivalent to *1(element).
([element] means 0 or 1 element.)

"string" or 'string'
matches the literal string given inside double quotes.

Other notations used in the productions are:

; or /* ... */
comment.

Appendix 7: P3P Guiding Principles (Non-normative)

This appendix describes the intent of P3P development and recommends guidelines regarding the responsible use of P3P technology. An earlier version was published in the W3C Note "P3P Guiding Principles" (http://www.w3.org/TR/NOTE-P3P10-principles).

The Platform for Privacy Preferences Project (P3P) has been designed to be flexible and support a diverse set of user preferences, public policies, service provider polices, and applications. This flexibility will provide opportunities for using P3P in a wide variety of innovative ways that its designers had not imagined. The P3P Guiding Principles were created in order to: express the intentions of the members of the P3P Working Groups when designing this technology and suggest how P3P can be used most effectively in order to maximize privacy and user confidence and trust on the Web. In keeping with our goal of flexibility, this document does not place requirements upon any party. Rather, it makes recommendations about 1) what *should* be done to be consistent with the intentions of the P3P designers and 2) how to maximize user confidence in P3P implementations and Web services. P3P was intended to help protect privacy on the Web. We encourage the organizations, individuals, policy-makers and companies who use P3P to embrace the guiding principles in order to reach this goal.

Information Privacy

P3P has been designed to promote privacy and trust on the Web by enabling service providers to disclose their information practices, and enabling individuals to make informed decisions about the collection and use of their personal information. P3P user agents work on behalf of individuals to reach agreements with service providers about the collection and use of personal information. Trust is built upon the mutual understanding that each party will respect the agreement reached.

Service providers should preserve trust and protect privacy by applying relevant laws and principles of data protection and privacy to their information practices. The following is a list of privacy principles and guidelines that helped inform the development of P3P and may be useful to those who use P3P:

- CMA Code of Ethics & Standards of Practice: Protection of Personal Privacy
- 1981 Council of Europe Convention For the Protection of Individuals with Regard to Automatic Processing of Personal Data
- CSA--Q830-96 Model Code for the Protection of Personal Information
- Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data
- The DMA's Marketing Online Privacy Principles and Guidance and The DMA Guidelines for Ethical Business Practice
- OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data
- Online Privacy Alliance Guidelines for Online Privacy Policies

In addition, service providers and P3P implementers should recognize and address the special concerns surrounding children's privacy.

Notice and Communication

Service providers should provide timely and effective notices of their information practices, and user agents should provide effective tools for users to access these notices and make decisions based on them.

Service providers should:

- Communicate explicitly about data collection and use, expressing the purpose for which personal information is collected and the extent to which it may be shared.
- Use P3P privacy policies to communicate about all information they propose to collect through a Web interaction.
- Prominently post clear, human-readable privacy policies.

User agents should:

- Provide mechanisms for displaying a service's information practices to users.
- Provide users an option that allows them to easily preview and agree to or reject each transfer of personal information that the user agent facilitates.
- Not be configured by default to transfer personal information to a service provider without the user's consent.
- Inform users about the privacy-related options offered by the user agent.

Choice and Control

Users should be given the ability to make meaningful choices about the collection, use, and disclosure of personal information. Users should retain control over their personal information and decide the conditions under which they will share it.

Service providers should:

- Limit their requests to information necessary for fulfilling the level of service desired by the user. This will reduce user frustration, increase trust, and enable relationships with many users, including those who may wish to have an anonymous, pseudonymous, customized, or personalized relationship with the service.
- Obtain informed consent prior to the collection and use of personal information.
- Provide information about the ability to review and if appropriate correct personal information.

User agents should:

- Include configuration tools that allow users to customize their preferences.
- Allow users to import and customize P3P preferences from trusted parties.
- Present configuration options to users in a way that is neutral or biased towards privacy.
- Be usable without requiring the user to store user personal information as part of the installation or configuration process.

Fairness and Integrity

Service providers should treat users and their personal information with fairness and integrity. This is essential for protecting privacy and promoting trust.

Service providers should:

- Accurately represent their information practices in a clear and unambiguous manner -- never with the intention of misleading users.
- Use information only for the stated purpose and retain it only as long as necessary.
- Ensure that information is accurate, complete, and up-to-date.
- Disclose accountability and means for recourse.
- For as long as information is retained, continue to treat information according to the policy in effect when the information was collected, unless users give their informed consent to a new policy.

User agents should:

- Act only on behalf of the user according to the preferences specified by the user.
- Accurately represent the practices of the service provider.

Security

While P3P itself does not include security mechanisms, it is intended to be used in conjunction with security tools. Users' personal information should always be protected with reasonable security safeguards in keeping with the sensitivity of the information.

Service providers should:

- Provide mechanisms for protecting any personal information they collect.
- Use appropriate trusted protocols for the secure transmission of data.

User agents should:

- Provide mechanisms for protecting the personal information that users store in any data repositories maintained by the agent.
- Use appropriate trusted protocols for the secure transmission of data.
- Warn users when an insecure transport mechanism is being used.

Appendix 8: Working Group Contributors (Non-normative)

This specification was produced by the P3P Specification Working Group. The following individuals participated in the P3P Specification Working Group, chaired by Lorrie Cranor (AT&T): Mark Ackerman (University of California, Irvine), Margareta Björksten (Nokia), Eric Brunner (Engage), Joe Coco (Microsoft), Brooks Dobbs (DoubleClick), Rajeev Dujari (Microsoft), Matthias Enzmann (GMD), Patrick Feng (RPI), Aaron Goldfeder (Microsoft), Dan Jaye (Engage), Marit Koehntopp (Privacy Commission of Land Schleswig-Holstein, Germany), Yuichi Koike (NEC/W3C), Yusuke Koizumi (ENC), Daniel LaLiberte (Crystaliz), Marc Langheinrich (NEC/ETH Zurich), Daniel Lim (PrivacyBank), Ran Lotenberg (IDcide), Massimo Marchiori (W3C/MIT/UNIVE), Christine McKenna (Phone.com, Inc.), Mark Nottingham (Akamai), Paul Perry (Microsoft), Jules Polonetsky (DoubleClick), Martin Presler-Marshall (IBM), Joel Reidenberg (Fordham Law School), Dave Remy (Geotrust), Ari Schwartz (CDT), Noboru Shimizu (ENC), Rob Smibert (Jotter Technologies Inc.), Tri Tran (AvenueA), Mark Uhrmacher (DoubleClick), Danny Weitzner (W3C), Michael Wallent (Microsoft), Rigo Wenning (W3C), Betty Whitaker (NCR), Allen Wyke (Engage), Kevin Yen (Netscape), Sam Yen (Citigroup), Alan Zausner (American Express).

The P3P Specification Working Group inherited a large part of the specification from previous P3P Working Groups. The Working Group would like to acknowledge the contributions of the members of these previous groups (affiliations shown are the members' affiliations at the time of their participation in each Working Group).

The P3P Implementation and Deployment Working Group, chaired by Rolf Nelson (W3C) and Marc Langheinrich (NEC/ETH Zurich): Mark Ackerman (University of California, Irvine), Rob Barrett (IBM), Joe Coco (Microsoft), Lorrie Cranor (AT&T), Massimo Marchiori (W3C/MIT), Gabe Montero (IBM), Stephen Morse (Netscape), Paul Perry (Microsoft), Ari Schwartz (CDT), Gabriel Speyer (Citibank), Betty Whitaker (NCR).

The P3P Syntax Working Group, chaired by Steve Lucas (Matchlogic): Lorrie Cranor (AT&T), Melissa Dunn (Microsoft), Daniel Jaye (Engage Technologies), Massimo Marchiori (W3C/MIT), Maclen Marvit (Narrowline), Max Metral (Firefly), Paul Perry (Firefly), Martin Presler-Marshall (IBM), Drummond Reed (Intermind), Joseph Reagle (W3C).

The P3P Vocabulary Harmonization Working Group, chaired by Joseph Reagle (W3C): Liz Blumenfeld (America Online), Ann Cavoukian (Information and Privacy Commission/Ontario), Scott Chalfant (Matchlogic), Lorrie Cranor (AT&T), Jim Crowe (Direct Marketing Association), Josef Dietl (W3C), David Duncan (Information and Privacy Commission/Ontario), Melissa Dunn (Microsoft), Patricica Faley (Direct Marketing Association), Marit Köhntopp (Privacy Commissioner of Schleswig-Holstein, Germany), Tony Lam (Hong Kong Privacy Commissioner's Office), Tara Lemmey (Narrowline), Jill Lesser (America Online), Steve Lucas (Matchlogic), Deirdre Mulligan (Center for Democracy and Technology), Nick Platten (Data Protection Consultant, formerly of DG XV, European Commission), Ari Schwartz (Center for Democracy and Technology), Jonathan Stark (TRUSTe).

The P3P Protocols and Data Transport Working Group, chaired by Yves Leroux (Digital): Lorrie Cranor (AT&T), Philip DesAutels (Matchlogic), Melissa Dunn (Microsoft), Peter Heymann (Intermind), Tatsuo Itabashi (Sony), Dan Jaye (Engage), Steve Lucas (Matchlogic), Jim Miller (W3C), Michael Myers (VeriSign), Paul Perry (FireFly), Martin Presler-Marshall (IBM), Joseph Reagle (W3C), Drummond Reed (Intermind), Craig Vodnik (Pencom Web Worlds).

The P3P Vocabulary Working Group, chaired by Lorrie Cranor (AT&T): Mark Ackerman (W3C), Philip DesAutels (W3C), Melissa Dunn (Microsoft), Joseph Reagle (W3C), Upendra Shardanand (Firefly).

The P3P Architecture Working Group, chaired by Martin Presler-Marshall (IBM): Mark Ackerman (W3C), Lorrie Cranor (AT&T), Philip DesAutels (W3C), Melissa Dunn (Microsoft), Joseph Reagle (W3C).

Finally, Appendix 7 is drawn from the W3C Note "P3P Guiding Principles", whose signatories are: Azer Bestavros (Bowne Internet Solutions), Ann Cavoukian (Information and Privacy Commission Ontario Canada), Lorrie Faith Cranor (AT&T Labs-Research), Josef Dietl (W3C), Daniel Jaye (Engage Technologies), Marit Köhntopp (Land Schleswig-Holstein), Tara Lemmey (Narrowline; TrustE), Steven Lucas (MatchLogic), Massimo Marchiori (W3C/MIT), Dave Marvit (Fujitsu Labs), Maclen Marvit (Narrowline Inc.), Yossi Matias (Tel Aviv University), James S. Miller (MIT), Deirdre Mulligan (Center for Democracy and Technology), Joseph Reagle (W3C), Drummond Reed (Intermind), Lawrence C. Stewart (Open Market, Inc.).