# Preferences Registry Format 1.0 Specification

A. Sosnin

11th May 2004

# Contents

# 1 Preamble

**Abstract**

Preferences Registry Format (is short: "PRF") is an XML-based simple configuration file format meant to be used in software that needs a simple, yet powerful, platform-independent way of storing configuration data. It is based on the simple name-value principle commonly used in many software configuration formats.

This format is meant to be generic and multi-purpose, but it is not designed to be universal. If you need a more complex, or a more specialised way of storing application configuration data, it is suggested to refer to a more comprehensive preferences file format named "CC/PP" which stands for Composite Capabilities/Preference Profiles [3]. The Preferences Registry Format is a major simplification of the ideas put into the CC/PP Resource Description Framework [4] based file format.

## 1.1 Status of this document

This document is a "request for comments" (RFC) draft document. It is a work in progress. Comments about any part of this document are welcome.

The design solutions, expressed in this document, are not yet intended for production use.

This document is subject to the copyright notice below.

## 1.2 Copyright notice

# 2 Preferences Registry Format 1.0

## 2.1 Well-formedness and conformance

It is required for the PRF file to be a well-formed XML file and fully conform to the following specification. Validation of PRF files can be performed using the DTD in section 3.2 (non-normative). For validation it is required that a PRF file contains the following line before the `PRF` root element:

```
<!DOCTYPE prf "/dtd/prf-1.0.dtd">
```

or, optionally, with a public ID:

```
<!DOCTYPE prf PUBLIC "-//tomatensaft//DTD prf 1.0 -//EN"
"http://tomato.dyn.ee/dtd/prf-1.0.dtd">
```

Note that the public ID is optional and may be ommitted. The system ID is better to point to a real DTD location, so that a validating DOM library could validate this file.

Also note, that, as defined in the XML 1.0 Recommendation [1], a common minimal XML parsing directive is required in the beginning of a PRF file before the document type declaration, e.g.:

```
<?xml version="1.0"?>
```

Optionally, there may be a declaration of the encoding, used in the file, using the `encoding` attribute. It defaults to "UTF-8", which is a recommended PRF file encoding.

## 2.2 Profiles

PRF supports multiple configuration profiles, which are delimited by the `profile` element. Each profile must have a unique ID, which is set with the `id` attribute. A profile may have an optional name. It is set using the `name` attribute. `profile` elements can only contain `component` elements inside.

Applications must use the `id` attribute to select needed `profile` elements. The `name` attribute is meant for a descriptive name specification (e.g. storing the user-selected name of a profile).

## 2.3 Components

The `component` elements are meant to logically stucturise PRF files. It is recommended to divide different configuration parts into different logical "component" sections, because it simplifies configuration file maintenance and software debugging.

Each component must have a unique ID defined with the `id` attribute. Please note, that it must be unique even between different profiles, so it is recommended to give their ID an appropriate value (e.g. "HTTPUpload1", "HTTPUpload2" etc.).

A component element can have an optional name attribute that can describe the nature of the component (e.g. "FTPUpload", if it is connected to FTP-based file uploading). Applications may use this field to find needed components with needed configuration options. The name field isn't meant for descriptive name specification, although the name may be somewhat descriptive.

### 2.4 Options

Options define name-value pairs of an application's configuration state. Each `option` element must have a name (NMTOKEN, alpha-numerical) associated with it. Note that in `option` element this field is mandatory. The value of each option is written inside the option element as PCDATA (parsed character data).

### 2.5 Limitations

Note that this specification does not define any "list" or other complex data type support. The users of this format may enhance the format by providing additional `option` content specifications. This format's `option` element is not, though, limited to contain simple text data and may contain embedded structured data, such as XML.

## 3 Appendix

### 3.1 Example

The following is an example of a PRF configuration file:

```
<?xml version="1.0"?>
<!DOCTYPE prf "dtd/prf-1.0.dtd">
<prf>
    <profile id="default1" name="default">
<component id="c1" name="Setup">
    <option name="ExecDir">/home/tomato/tale2c</option>
    <option name="ConfDir">conf</option>
</component>

     <component id="c2" name="HTTPUpload">
         <option name="ServerHostName">zzx.dyn.ee</option>
         <option name="ServerPortNumber">80</option>
         <option name="ScriptName">/edit.php</option>
</component>

<component id="c3" name="FTPUpload">
         <option name="ServerHostName">album.dyn.ee</option>
         <option name="ServerPortNumber">21</option>
         <option name="UploadDir">/tale2/albums/</option>
</component>

<component id="c4" name="DocumentSetup">
         <option name="DefaultFilename">Untitled.xml</option>
     </component>
    </profile>
</prf>
```

### 3.2 Document Type Definition

The following code is a non-normative DTD, useful for Preferences Registry Format files validation:

```
<!ELEMENT prf (profile*)>

<!ATTLIST profile
xmlns CDATA #IMPLIED >

<!ELEMENT profile (component*)>

<!ATTLIST profile
id ID #REQUIRED
name CDATA #IMPLIED >

<!ELEMENT component (option*)>

<!ATTLIST component
id ID #REQUIRED
name CDATA #IMPLIED >

<!ELEMENT option (#PCDATA)*>

<!ATTLIST option
name NMTOKEN #REQUIRED >
```

# References

### 3.3 Normative references

[1] Extensible Markup Language 1.0, Second Edition, W3C Recommendation 2 October 2000, *http://www.w3.org/TR/REC-xml*

### 3.4 Informative references

[2] PRF 1.0 API, Specification and Implementation Notes *DRAFT*, *http://zzx.dyn.ee/pub/doc/free/prf1-api/libprf1.pdf*

[3] CC/PP (Composite Capabilities/Preference Profiles): Structure and Vocabularies 1.0, W3C Proposed Recommendation 15 October 2003, *http://www.w3c.org/TR/2003/PR-CCPP-struct-vocab-20031015/*

[4] Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, *http://www.w3.org/TR/REC-rdf-syntax/*