

# OLD PAGE BEFORE CLEANUP

THIS IS AN OLD ARCHIVED PAGE -- NO LONGER ACCURATE

## Discussion on Profile Structure

Technical goals (see [Goals of New Version of Standard|display/ISO24751/Goals+of+New+Version+of+Standard]):

- Allow for continuous updating of needs and preference terms
  - Define a process for maintaining a core set of needs and preference terms.
  - Define a process for adding Additional core and other terms
  - Define a process for adding (non-core) device/application specific properties
  - Allow for inclusion of **USE/CONTAINER** - though we need to figure out what all the ways or meanings of context are. Is it a continuum or are there different types of **USE/CONTAINER**?
    - Part of the product that the preference applies to (e.g. font size for text, or headers, or footnotes, or captions or...)
    - Device/Platform
    - User situation
    - "All the properties that determine the specific setting that should be used."
      - time of day
      - fatigue level
      - physical environment
      - part of the product the setting will be applied to
      - person you are interacting with
    - [LN] context properties have been suggested many times - they often seem to be ones that help identify the relevant set of needs and prefs, whereas 'device-related' properties can be either like context or specific needs. I think we should decide if we mean that 'context' can be a determinant of needs (identifying a set) or if we want to describe it for some reason???
  - Allow for generic and individualized profiles to be used for user interface and digital resource adaptation
    - [LN] We worked on this in 25751 and I think did a good job - we had needs and prefs of a generic kind and then some that related to the interface/assistive tech - eg I am using Jaws so there are some things I need for that which only work in Jaws.
  - Ability to have generic and very individualized contexts -\*\* [LN] does this mean two sorts of use cases: a needs and prefs profile for everyone sitting in a lecture theatre and then one for me with low vision? do we foolow the WCAG/CSS lesson of having the individual one overriding the generic one?
  - **USE/CONTAINER**-sensitive and non-**USE/CONTAINER**-sensitive preferences -
    - [LN] isn't this already covered with whatever we do with contact etc?
  - Properties may apply to a specific user interface platform or be generic (cross-platform)
    - [LN] ditto
  - Users should have an easy way of modifying their profile, and should not be required to understand the raw profile
    - [LN] of course!.
1. "USE/CONTAINER?" < PART## COMMON (USE/CONTAINER?) for adapting a profile (or for switching to a different one?)
    - a. i. **device (that is being adapted)** (
      - ii. **Phone vs computer**
      - iii. **contextually responsive**
      - iv. **environmental**)
  2. **User specific Conditions for switching profiles**
    - a. (when I am in my AMIGO vs when I am at my desk)

## Proposed key points for discussion:

#	Proposed key point for discussion	Agreed
1	The registry of user preferences consists of property-value pairs in a flat structure. (Discussion)	
2	Are the user profiles themselves all flat or are they layered? (Discussion)	
3	Property names are URIs, values are of any type that can be stored as a string. (Discussion)	
4	Should we have URLs rather than the more general URI? (Discussion)	
5	There are two categories in the registry: core and non-core. (Discussion)	
6	How to refer to main registry and application specific registry? (Discussion)	
7	What do we call the items in the Registry? (Discussion)	
8	How to handle items that are duplicates: Identical meaning but a different name? (Discussion)	
9	Does a key definition consist of its URI and its type and value space? (Discussion)	
10	In a user preference profile instance, the value of those keys that are not present, are unknown (incomplete profile). (Discussion)	

11	In a user preference profile instance, one key may occur multiple times, but only with different values. <a href="#">(Discussion)</a>	
12	Do we need tags attached to keys, for example to link eMail addresses to users or specify languages? <a href="#">(Discussion)</a>	
13	In a user preference profile instance, each entry (property-value pair) may have a probability assigned. <a href="#">(Discussion)</a>	
14	Where other standards provide useful key definitions, can we import them by using their domain as part of the key URI? <a href="#">(Discussion)</a>	
15	We will not develop key definitions related to device and usage context. However, should we import such keys from other standards? <a href="#">(Discussion)</a>	
16	Are the core properties the key definitions that have been accepted by ISO 24751? <a href="#">(Discussion)</a>	
17	Are the non-core properties the key definitions that have been provided by vendors, user groups, other standards groups, or any third parties? <a href="#">(Discussion)</a>	
18	Are both core and non-core properties stored in the registry database? <a href="#">(Discussion)</a>	
19	Is the registry database hosted by Raising the Floor under reg.gpii.net? <a href="#">(Discussion)</a>	
20	Should we have a web-based interface to provide access to the registry for the public? <a href="#">(Discussion)</a>	
21	Does a part of ISO/IEC 24751 define the data model of the registry? <a href="#">(Discussion)</a>	
22	Does a part of ISO/EIC 24751 define the process of adopting core and non-core properties into the registry on a regular basis? <a href="#">(Discussion)</a>	
23	Should views may be specified on top of the key-value pairs, each defining a particular structure/ontology for the key-value pairs? <a href="#">(Discussion)</a>	
24	Are views specified externally to the registry (e.g. as RDF/OWL files on a Web server)? <a href="#">(Discussion)</a>	
25	What is our preferred practice for finding or adding preferences? <a href="#">(Discussion)</a>	
26	How do we represent different settings/variations for different contexts (e.g. times, ambient light, locations)? <a href="#">(Discussion)</a>	

## Discussion

- [NEW - AGREED]** The registry of user preferences consists of property-value pairs in a flat structure.
- [OPEN]** Are the user profiles themselves all flat -- or are they layered?????#\* Some scenarios may require a more complex structure. Example: "Visual content requires text alternatives". Can be flattened by multiplying out the subject or both subject and object. "visual-content-requires-text" or "visual-content-requires-text: true". Better: "Text-alternative-for-visual-content: true".
  - Pragmatic approach, no semantic parsing required.
  - This is not meant to be exposed to the end user.
  - Goal to make this most shareable/interoperable as possible.
  - Should not have complex structures - we need simple structures that could be combined to express complex things
  - Still need to consider Semantic Web concepts if they are applicable to our case. RDF and tool support. Erlend will post some information on the Wiki. But how much of this has been accepted in the user profile area? Need to get vendor support.
  - Two sides of the coin: User preference profile = simple structure of property-value pairs; user model = can be sophisticated ontology /structure expressing relationships etc
- [AGREED]** Property names are URIs, values are of any type that can be stored as a string. #\* Rationale: Link to semantic Web via URI, extensible system
  - For matching algorithms there is only a flat set of properties.
- [OPEN]** Should we have URLs rather than the more general URI?
  - Two purposes for using URIs:
    - First - uniqueness of keys.
    - Second - reference to description of key (in ontology or similar). (do we need to have them be a URL for this?)
  - If URIs can all be found on the Web -- then no need to limit to URL.
- [AGREED]** 2 Layers to the Registry:
- [AGREED]** What should we name the two categories#\* names for category 1 - the accepted recommended terms#\* **core, recommended, preferred, # #\* #\* [AGREED] ON CORE#** names for category 2 - New unscreened terms#\* **live, new, candidate, no name (just not recommended)#\* #\*\*\* [AGREED] ON NON-CORE**
- [OPEN]** How to refer to MAIN Registry and APPLICATION SPECIFIC
- [OPEN]** What do we call the items in the Registry## Candidates#### Property-Names#### Property [AH Proposal - call them Preference Definitions - then we can have Preference Definition Name, Preference Definition Value, Preference Name, Preference Definition Reference, Preference Value (the last three being in instances). Rationale: "Property" is well overloaded with many different confusing meanings and also geeky for non-IT people]### Need-Preference
- #\* [LN] I am not sure why we have this 'live' term?? I don't understand what it means. Does it mean user chosen/local...new??? If people have new terms they find useful and register them, they should be permanent????#\* [GV] They are live until they are examined and an editorial group declares them CORE. Since anyone can put something in (and it might be a duplicate or not well named or thought out) we have LIVE to all anyone to enter and CORE to recognize things that are more permanent. All LIVE can become CORE - and should when they are proven or vetted.#\* [LN] why note just have terms and if we decide they are 'core' then we can designate them 'core' I suspect we'd do this in a how-to?#\* Unregistered properties can be used by any vendor (e.g. application-specific settings and data blocks).
  - The API must pass both registered and unregistered properties. One API or multiple APIs?
    - [LN] We need to be sure how the records are exchanged to maximise their interoperability (below Gregg talks of a block - I think it is more conventionally called a record.
  - Core properties start with a defined domain name, e.g. <http://reg.gpii.net/<ns>/<userpref#>>. Live and vendor-specific properties have a proprietary domain name or may have a namespace at [reg.gpii.org](http://reg.gpii.org).

- We also allow (but don't encourage) the storing of application-specific settings as block. E.g. all TextHelp settings in a bundle. "Virtual block" of many properties. ~~with common domain name, e.g. <http://texthelp.com/ns/userprof>. This can be handled as a single name-value pair in a vendors (or other) location. - This is about adaptation settings - different from properties [??] in the user profile. But with a 1:1 matching, there could be counterparts of settings in a user profile.~~ While this is allowed, it is preferred for vendors to store their user preferences using common names from the registry that match their use in the application. [Edits by GV## [LN] I think we should borrow from practice where we have some properties and people can make more and then they use a set that they choose, or perhaps one that is recommended. The sets are known by DCMI as 'Application Profiles' - sorry it's not a good nam but I think we could find a name. The development of 'sets' of properties often happens because of local implementations, or groups agreeing they will use a certain set, etc.
  - User preference profile storing expected behavior of applications only or describing persons? - Statistical methods don't care. What does this mean please?
  - We need to get the big vendors on it. This is a way to achieve this. But what if the vendor doesn't want to tell what their application can tweak (in terms of adaptation parameters)?
    - [GV] In these cases the vendor can use the single name-value pair data block to store their preferences (or any portion they want to keep secret). Even here though we recommend that they store all that they can using the common preferences and use the data block only for those that there is a special reason to.
  - What if a new technology comes along? Live space can host new properties, so that others know. Approval committee will look at properties in live part, and promote it to core (optionally renaming it) where appropriate. yep - this is above...
  - Culture is a context.
    - [LN] very interesting - who determines my culture?
4. [OPEN] How to handle items that are duplicates - identical meaning but different NAME#\* put them both in Registry.
    - ##\* Choose one as the RECOMMENDED
      - Put a pointer in the VALUE? DEFINITION? that points to URL of RECOMMENDED NAME
  3. [OPEN] The definition of a key consists of its URI, and its type and value space (e.g. enumerated values, integer). (See <http://myurc.org/TR/res-prop-vocab1.0/> as an example.)
    - [GV] a number - rather than a name is used for the unique ID although both would be provided in the URI for human reference (e.g. [reg.gpii.net/core/34200\(name\)##](http://reg.gpii.net/core/34200(name)##) In addition to facilitating internationalization - it also allows a preference to continue working even if the name is changed. ##\* (when presented to humans, the number (or the whole URI) can be replaced with the NAME in the language of the user. ##\* [GV] Core and live share the same namespace. Otherwise promoting something from live to core will break all past uses. ##\* [GV] CORE would be a flag in the registry rather than a different namespace. and the example would be [reg.gpii.net/common/34200\(name\)](http://reg.gpii.net/common/34200(name)))
    - ##\* [LN] What is the needs for this if we are using namespaces and if we allow for/expect people to use local translations of property names in their wizards etc? eg a Russian version but it still refers to the original namespace?##\* [GV] See below. if we edit the name -- it would break all of the existing profiles. the number would be the unique id - not the name.
    - For internationalization the NAME and DESCRIPTION of each key is stored separately. I recommend using the ISO template we set up for Metadata for Learning Resources...
    - GZ: Why using a number for the sake of internationalization? It is better to use an English term (that is still human readable) than using an arbitrary number with no reminiscense to the topic. English terms are equally suited for internationalization, but easier and less error-prone for developers. Do you really like those generated URIs of the form "http://example.com/?page=1434"?
    - [GV] I was suggesting that the NUMBER be the Unique ID. That way, if we find that we have chosen the wrong name -- or have to add a word to it to make it more specific (e.g. "volume" is changed to "volume-main") we don't break all the profiles out there.
    - **BUT I have thought about this for a bit more - and i now think that we should have BOTH a NUMBER and an ENGLISH NAME for each item. People read the NAME and machines read the NUMBER. That way it is always human readable and yet we can change the name if we have to. Our tools can always update the name to the latest form each time they touch the profile. And people from different countries can always have the name displayed in their language if the choose when they view the registry. in fact we can have parallel copies of the registry in different languages (tied together by the number). The ENGLISH language registry however would be the master. The others for convenience - since it will contain both the NAME and the DESCRIPTION in their language.**
    - NOTE: There may never be other language versions-- it may be that all people who do profiles will always speak english well. But i don't think we should assume that -- and should make it easy for a machine to process a profile that is in another language without having to do language translation on it.
    - i have changed my edits above to add name so it would be NUMBER(name) with the NUMBER being the UniquePermanentID for the item.
    - GZ: Having a number and a name is worse than just one of them. We make the URI longer, and introduce more sources for errors. If either the number or the name is changed this means a different URI and thus a different property. For example, [http://reg.gpii.net/common/34200\(volume\)](http://reg.gpii.net/common/34200(volume)) is a different property from [http://reg.gpii.net/common/34200\(main-volume\)](http://reg.gpii.net/common/34200(main-volume)). We don't need a number for machine readability - machines can read names just as well as numbers. But for humans words are easier to read, to remember and less error-prone. I claim that <http://reg.gpii.net/common/volume> is better than <http://reg.gpii.net/common/34200>. In any case, we will define human-readable labels for each property URI in any language - even for the English language. Thus we can change the English label even after the URI has been specified. Note that this doesn't mean that we have different registries for different countries. We have only one registry, and in there we define the property URIs and for each human-readable labels for many languages, including English. If you look at any of the standards that we have looked at so far, they are all using names in some or the other way to define properties, rather than plain numbers. Examples from Dublin Core and ETSI: <http://purl.org/dc/elements/1.1/title>, [<http://purl.org/dc/terms/conformsTo>, <http://uri.etsi.org/upm/interaction-preferences/interaction-timeout>, <http://uri.etsi.org/upm/interaction-preferences/preferred-output-modality> (and i stongly suggest we just adopt their URIs into our registry).
    - [LN] I agree that the URI identifies the definition but again, recommend a way of doing the definition and, specifically, the table we have for ISO 19788
  4. [AGREED] In a user preference profile instance, the value of those keys that are not present, are unknown (incomplete profile).
    - [GV] Not present where? at the URI?
    - [LN] If we do not require properties to have values, we don't need to worry about incomplete profile sets...
  5. [AGREED] In a user preference profile instance, one key may occur multiple times, but only with different values.
    - [GV] What does this mean? in a registry or a preference profile? which should be used?
    - [LN] In the registry, each property should be defined as a pair etc but the same property may have two values eg language property might have both English and Readability Level 2 as values so it would occur twice (depending on what values are expected for 'language' of course).
  6. [OPEN] In a user preference profile instance, some keys may have a language tag attached to their value for disambiguation in case of multiple occurrences in a profile. ## [GV] What does this mean?
  7. ##\* Like a second key layer (but not used as official key).
    - Other use cases? E.g. home vs. office phone number? - But you can accommodate for that by using different property names.

- [LN] Suppose there is an English description of an article and also a French description, or there are two authors and their email addresses need to be associated with the correct name, ...
8. **[AGREED]** In a user [preference](#) profile instance, each entry (property-value pair) may have a probability assigned. The default value is 1 (100%).
    - [LN] Is this to do with need vs preference? In 24751 we distinguished between 'have to have' - needs and 'prefer' - as prefs. I think this may be better.
    - GZ: This is not priority - this has to do with incomplete user profiles being filled by automatic tools. They may fill the gaps, but - of course - these are just guesses with a certain level of probability.
    - Call 2012-02-28: Jutta: Another feature where probability is involved is "partially needs to meet", "completely needs to meet" (see 24751). How well the resource meets a particular user need. But that's related to DRD.
  9. **[OPEN]** Where other standards provide useful key definitions, we can import them by using their domain as part of the key URI (e.g. "<http://purl.org/dc/elements/1.1/title>").
    - [GV] (this is always allowed but is it a good idea for stability? Should this be an option (it always is) or recommended. (If borrowed then we should credit in the description)
    - [LN] Do you mean import them as core terms or something or do you mean include them in a record? After working with such things for a while in DCMI, both the ed group and the accessibility group have recommended having their properties as modules of properties to be used with others - leaving other properties where they are ...
    - GZ: Agree. Just including as records.
  10. **[OPEN]** We will not develop key definitions related to device and usage context. However, we may import such keys from other standards (e.g. from OMA UAProf or W3C Delivery Context Ontology).
    - [LN] This is done easily by having the set identified (the schema registered) but not having all the propertie definitions in the same place.
  11. **[OPEN]** The "core properties" are key definitions that have been accepted by ISO 24751.
    - [GV] I thought 24751 was being revised to not include ANY properties. Instead we were moving to a registry model. 24751 would define structure and process but not include any specific values. we can use current 24751 properties as a starting point
    - [LN] I agree that we use what definitions are good for us and re-define them in a registry...
  12. **[OPEN]** The "live properties" are key definitions that have been provided by vendors, user groups, other standards groups, or any third parties.
  13. **[OPEN]** Both core properties and live properties are stored in a registry database.
  14. **[OPEN]** The registry database is hosted by Raising the Floor
    - [GV] under the gpil domain -- reg.gpil.net.
  15. **[OPEN]** A Web-based interface provides access to the registry for the public.
  16. **[OPEN]** A part of ISO/IEC 24751 defines the data model of the registry.
  17. **[OPEN]** A part of ISO/EIC 24751 defines the process of adopting core and live properties into the registry on a regular basis.
  18. **[OPEN]** "Views" may be specified on top of the key-value pairs, each defining a particular structure/ontology for the key-value pairs.
  19. **[OPEN]** Views are specified externally to the registry (e.g. as RDF/OWL files on a Web server).
  20. **[OPEN]** [GV] IN USE the preferred practice is:## use a common preference (core or live) if one exists that fits## if none applicable - then ADD a new live preference## for application-specific settings (e.g. location of the toolbar on top or left of xyz view in qrs application) use a vendor or product specific namespace. ## only where necessary use a data block to store preferences (and this would be in a application-specific namespace## NOTE: reg.gpil.net will provide space for people / organizations to maintain vendor or product specific namespaces. This will be especially useful for open-source product but can be used by any vendor that does not want to maintain their own preference server.

## Proposed Definitions

**COMMON** - These are preferences that are generic in nature and would be used by multiple applications. They are intended to be defined and used in common between applications. (They would stored in the MAIN REGISTRY)

There are **two types of COMMON preferences**

**CORE (COMMON):** These are COMMON preferences that have been studied and are believed to be stable and fixed. A committee will review the LIVE COMMON (or LIVE) preferences and determine when they should become CORE COMMON (or CORE). In tagging them with CORE designation, the name or definition or value range might be tweaked but care must be take to not break any use of the tag by existing user preference profiles.

**LIVE (COMMON):** These are COMMON preferences that have been entered by someone because there was not already another COMMON preference that met the need. Anyone can submit a new preference to this set and it will be included unless it is an obvious duplication. All LIVE preferences are candidates to become CORE.

These two categories of COMMON preferences would be maintained in the MAIN REGISTRY.

**APPLICATION SPECIFIC** - These are settings that would never be candidates for a COMMON preference because they are so specific to an application that they would not make sense as a COMMON preference. An example may be whether a particular toolbar in an application is located above or below another toolbar in the application or off to the left.

APPLICATION SPECIFIC preferences are maintained by whomever is maintaining the application -- and would not be in the MAIN REGISTRY. (Preferences always include a pointer to which registry they are defined in)

**USER PROFILE MODEL** - The structure of a user preference profile.

**USER PROFILE INSTANCE** - A user preference profile with data for a specific person or group of persons.

## See Also

See also:

- Gottfried's [User Profile Illustration](#)
- Liddy's [Thoughts on New Work](#)
- Gottfried's [Statements for user profile standard development \(for discussion\)](#)
- An example [Scenario with Conditional Items in A User Profile](#)