Network Working Group
Document: draft-ietf-acap-dataset-model-00.txt
Expires August 21, 1999

Ryan Troll
Carnegie Mellon
February 16, 1999

An Introduction to the ACAP Dataset Model

<draft-ietf-acap-dataset-model-00.txt>

Status of this Memo

Abstract

The ACAP [ACAP] Dataset Model is very extensible, and allows applications to easily share options and information. With this extensibility comes a complexity that an application designer must fully understand in order to interoperate while using ACAP.

This document will help the reader understand and visualize the ACAP hierarchy, come to a better understanding of how to design and access ACAP datasets, and understand the relationship between attributes, entries, datasets, and dataset classes.

1. Introduction

The Application Configuration Access Protocol [ACAP] is designed to support remote storage and access of program option, configuration, and preference information. The data store model is designed to allow a client relatively simple access to data, to allow new information to be easily added without server reconfiguration, and to promote the use of both standardized data and custom or proprietary data.

This document explores the data store model, allowing the reader to visualize what the ACAP data store actually looks like. With this picture, developers will be able to design an ACAP dataset class for their own applications, and interoperate with others.

1.1. Conventions Used in the Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [KEYWORDS].

1.2. Terminology

Dataset Class — A dataset class is a formal description of how to use information stored within a dataset. Descriptions of all attributes, and what their values mean, make up a dataset class.

Dataset — Every level of the ACAP hierarchy is a dataset. Datasets contain entries, and/or other datasets. Datasets may inherit entries from other datasets, or be standalone repositories of information.

Sub-Dataset — A dataset may contain pointers to other datasets. These datasets are subsets of the main dataset, and may contain entries that are grouped together for a common reason. In a dictionary, words are grouped together based on their first letter. These smaller groupings are subsets of the entire dictionary, just like sub-datasets are subsets of the entire dataset.

Sub-datasets do not have to reside on the same ACAP server as the dataset containing the sub-dataset pointer. A site may provide an ACAP server for all users, and this ACAP server may contain custom bookmarks for all users. If a user decides to use another ACAP server for his/her ACAP needs, one entry in the users bookmark dataset could be a sub-dataset pointer back to the site-wide dataset.

A full dataset may also be a sub-dataset of another dataset. Being a sub-dataset does not change how the dataset is accessed or viewed. Sub-datasets provide a way for a dataset to break up its data into smaller chunks, and/or share data between datasets.

Entry — One item in a dataset is an entry (IE: A web

bookmark).
An entry consists of a collection of attributes.

Attribute — The smallest piece of information in an ACAP dataset. An attribute describes one aspect of an entry, and consist of a name and a value. (IE: the name 'color' and the value 'black')

2. ACAP Data Model

This section defines all of the parts of the ACAP data model, starting with the smallest piece of information.

2.1. Attribute / Value

The smallest piece of information in the ACAP data store is an attribute. An attribute contains two pieces of information: a name, and a value. Attributes may also have multiple values. Multiple values are stored as parenthesized lists of values. The value types Atom, Number, String, Parenthesized List, and NIL are defined in [ACAP], section 2.6.

Some attributes are already defined, and are used for every entry. These attributes include "entry", which is the name of the entry, and "subdataset", which is used to indicate that the entry represents a sub-dataset.

The bookmark dataset [BOOK] includes the attributes "bookmarks.Description", containing a user-assigned description ("CMU's ACAP Page"); "bookmarks.Name", containing the name of the reference ("ACAP Home Page"); and "bookmarks.URL", which is the bookmarked URL ("http://asg.web.cmu.edu/acap"). These attributes, along with others, will allow clients to use and/or present the bookmarks to the user.

The dictionary dataset [DICT] includes the attributes "entry" and "dictionary.definition". The attribute "entry" may have the value "spam", while the attribute "dictionary.definition" may have the value "("To send multiple unsolicited messages" "A meat-like substance")".

2.1.1. Attribute Naming Scheme

Attributes MUST adhere to a strict naming scheme. Attributes names which do not contain a dot (".") are reserved for standardized attributes which have meaning in any dataset. ("entry" and "modtime", for example.) All other attributes MUST have a name of the format "Dataset.Attr", where 'Dataset' is the name of the

dataset, and 'Attr' is the name of the attribute within the dataset. ('Attr' may contain dots.)

Some attributes are pre-defined, and have special meanings. These attributes are:

entry — The name of the entry within the dataset.

modtime — Timestamp of the last modification to anything within this entry.

subdataset — If set, this attributes indicates the existence of a sub-dataset of this entry.

For more information on the predefined attributes, consult [ACAP], section 3.1.1.

Attribute names are unique across all datasets. This is accomplished by the attribute naming scheme: all attribute names MUST be of the form "<dataset class>.<attribute-name>".

Within the dictionary dataset, all entries contain the attribute "dictionary.definition", which is a list attribute containing the definition(s) of the word. Also, the attribute "entry" takes on an additional meaning in the dictionary dataset: it is the word the entry is for.

2.1.2. Searching for Specific Attributes

When searching an ACAP dataset, a comparison function (referred to as a search comparator) must be specified, in order to handle the different types of values. When searching for a specific string, the "i;ascii-casemap" may be used, while when searching for a number the "i;octet" comparator may be used. For more information on what comparators are available, refer to [ACAP], section 3.2.

2.2. Entry

An entry is a collection of attributes, grouped together to describe a common item. Entries within a dataset MUST have a unique name. The design of the ACAP dataset MUST make sure this is the case. The entry name does not have to have any bearing on the actual entry content. (For an example of this, see the example at the end of this document.)

It is up to the client to choose a unique name for any entries it is going to create in the ACAP hierarchy. At this time, there is no standard way to choose an entry name.

For example, when using ACAP to store a personal dictionary, the entry name may be the word. Since the word may have multiple definitions, the dictionary dataset has been structured to use multi-valued attributes to allow clients to store multiple definitions.

On the other hand, an ACAP entry describing a bookmark may contain the URL, name, type, and last visited time. The name of the bookmark entry may be a numeric uniquifier, which has no meaning to the rest of the entry.

Another example is to compare the ACAP data store to a filesystem. In this case, every entry may be considered a directory entry. The contents of the files are irrelevant, and only the file attributes are examined. The name of the file is unique within the dataset, just like the dataset entry names, and all entries within the directory have an owner, a modification time, a last accessed time, and a size.

2.3. Dataset

A dataset is a collection of entries and datasets, which represent a set of the information being defined. When looking at a collection of bookmarks, a dataset may be defined for each of your folders of bookmarks, with a fifth dataset containing all of these folders, as well as any unfiled bookmarks.

2.3.1. Dataset Inheritance

Datasets may also inherit values from other datasets. Sites may want to define a base set of bookmarks that all of their users see. This can be accomplished by using dataset inheritance. When accessing a dataset that inherits from another, the accessed dataset appears to contain everything that is actually stored in it, as well as what is stored in the dataset it is inheriting from.

For example, a site may want to have a collection of Web bookmarks that are available for all users. If every user's bookmark dataset is configured to inherit from the master bookmark dataset, clients that fetch all of the user's bookmarks will not only see the user's personal bookmarks, but also the site specific bookmarks.

One important thing to remember: a client can not distinguish between inherited entries, and entries that are actually in the dataset, unless the search that is performed specificly says not to use inheritance. Designers and adminstrators MUST be careful when using inheritance.

2.4. Dataset Class

A dataset class is a formal definition of a dataset. The class definition describes what data will be stored, where the information will be stored, and what format it will be stored in. This allows multiple applications to use the same data within the ACAP server, without having to worry about breaking it for other applications. As long as all applications adhere to the dataset class definition, there will be no problems.

Standardizing dataset class definitions allows multiple applications to share data, even if they are from different vendors. By defining a common bookmark dataset class, all web browsers would be able to share their bookmarks with one another. In addition, by storing their bookmarks on an ACAP server, multiple instantiations of browsers across multiple computers and operating systems would be able to safely modify and update the list, with all other apps picking up the changes as they are made.

2.4.1. Dataset Class Naming Scheme

All dataset classes have a name. The name is used to locate the dataset class within the hierarchy. Dataset class names MUST either be of the form "vendor.<vendor/product>", or be specified in a standards track or IESG approved experimental RFC. The proposed dictionary dataset class will use the dataset class name "/dictionary", while a vendor's mechanism for storing application-specific options (such as window location) may be under "/cyrusoft.mulberry".

3. Hierarchy

The dataset namespace is a slash-separated hierarchy. The first part of the namespace is the dataset class. For standard datasets, this will be the name specified by the standard, such as "/bookmarks" or "/dictionary". For vendor-specific dataset classes, this will be the name of the vendor and the product, separated by a period. (IE: "/cyrusoft.mulberry")

The second part of the namespace is the scope of the dataset to be looked up. This can be "site", for server-wide datasets; "group" for administrative group datasets; "host" for host specific data; or "user" for a user's data.

If the scope is "group", "host", or "user", the third part of the namespace identifier is the name of the group, host, or user being looked up. (For example, "/cyrusoft.mulberry/host/lister.net.cmu.edu" would be Mulberry

preferences specific to the host "lister.net.cmu.edu".)

To fetch the bookmarks for user "ryan" on the ACAP server, the dataset to be searched would be "/bookmarks/user/ryan". If "ryan" is the current ACAP user, this may be abbreviated as "/bookmarks/~".

The dataset "/byowner" is reserved, and allows searches to show what dataset classes are owned by a given user. Searching "/byowner/user/ryan" will show all of the dataset classes in use by user "ryan".

Finally, searching the dataset "/" is equivalent to searching "/byowner/user/<current-user>". (Or "/byowner/~").

4. Bookmark Dataset Class

The ACAP bookmarks dataset class is defined in [BOOK]. The following example contains a couple of entries in this dataset, to demonstrate sub-datasets as well as give a visible example of what the dataset hierarchy looks like.

4.1. Example

The following example shows what data is in ACAP, and how that data may be used to generate a nice list of web bookmarks for your browser.

4.1.1. ACAP Bookmark Dataset

In dataset /bookmarks/user/ryan

```
        entry                   104
        bookmarks.Name          "Lurkers Guide to Babylon 5"
        bookmarks.Description   "Cool site with very complete episode guides"
        bookmarks.URL           "http://www.midwinter.com/lurk"
        bookmarks.Type          "link"

        entry                   12dag8
        bookmarks.Name          "SSH (Secure Shell) Home Page"
        bookmarks.Description   ""
        bookmarks.URL           "http://www.cs.hut.fi/ssh/"
        bookmarks.Type          "link"

        entry                   13ksjhdfgpoa
        bookmarks.Type          "separator"

        entry                   14roblink
        subdataset              "//other.acap.domain//bookmarks/rob/public"
        bookmarks.Name          "Rob's Public Bookmarks"
        bookmarks.Type          "folder"

        entry                   15emailfoo
        subdataset              .
        bookmarks.Name          "Email Stuff"
        bookmarks.Type          "folder"

    In dataset /bookmarks/user/ryan/email-stuff

        entry                   2baz212
        bookmarks.Name          "ACAP Home Page"
        bookmarks.Description   "It's not LDAP"
        bookmarks.URL           "http://asg.web.cmu.edu/acap"
        bookmarks.Type          "link"

        entry                   3bar51
        bookmarks.Name          "IMAP Home Page"
        bookmarks.Description   "Protocol for fetching messages"
        bookmarks.URL           "http://asg.web.cmu.edu/cyrus"
        bookmarks.Type          "link"

        entry                   812foo
        bookmarks.Name          "Sieve Home Page"
        bookmarks.Description   "Sieve mail filtering language"
        bookmarks.URL           "http://asg.web.cmu.edu/sieve"
        bookmarks.Type          "link"
```

4.1.2. Generated Bookmarks File

Based on this information, the following web bookmark file can be generated:

```
Lurkers Guide to Babylon 5
SSH (Secure Shell) Home Page
----------------------------
Rob's Public Bookmarks ->
Email Stuff ->
```

And, the "Email Stuff" folder / sub-menu would contain:

```
ACAP Home Page
IMAP Home Page
Sieve Home Page
```

5. Copyright

6. References

[KEYWORDS] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997

<ftp://ds.internic.net/rfc/rfc2119.txt>

[ACAP] C. Newman, J. G. Myers, "Application Configuration Access Protocol", RFC 2244, November 1997

<ftp://ds.internic.net/rfc/rfc2244.txt>

[DICT] C. Daboo, "ACAP Personal Dictionary Dataset", Work In Progress, March 1998

[BOOK] R. Gellens, "ACAP Bookmarks Dataset Class", Work In Progress, December 1998

7. Author's Address

Ryan Troll
Computing Services
Carnegie Mellon
5000 Forbes Avenue
Pittsburgh, PA 15213

Phone: (412) 268-8691
EMail: ryan@andrew.cmu.edu