

INTERNET-DRAFT

[draft-ietf-uri-urn-handles-00.txt](#)

William Arms

David Ely

Corporation for National Research Initiatives

June 23, 1995

Expires: December 23, 1995

The Handle System

Status of this document

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

[Page 1]

Abstract

The Handle System provides identifiers for digital objects and other resources in distributed computer systems. These identifiers are known as handles. The system ensures that handles are unique and that they can be retained over long time periods. Since the system makes no assumptions about the characteristics of the items that are identified, handles can be used in a wide variety of systems and applications.

The handle system has the following components: naming authorities, handle generators, the global handle server, local handle servers, caching handle servers, client software libraries, proxy servers, and administrative tools. For reasons of performance and availability, the global, local, and caching servers are implemented as distributed systems comprising many server computers. All components, except the local handle server, have been implemented and are available for general use by the research community.

The handle system provides all the capabilities listed in RFC 1737, K. Sollins, L. Masinter, "Functional Requirements for Uniform Resource Names", 12/20/1994.

Contents

[1](#) Introduction

1.1 Overview

1.2 Historical background

1.3 The handle system and the Internet Engineering Task

Force

[2.](#) The handle system

2.1 System components

2.2 Policies and procedures

[3.](#) Basic Concepts

3.1 Handles and handle records

3.2 Resolution of handles

3.3 Indirect handle

3.4 Syntax

[4.](#) Naming Authorities

4.1 The naming authority hierarchy

4.2 Creating naming authorities

4.3 Primary handle server for a naming authority

4.4 Handles for naming authorities

[5.](#) Handle Servers

5.1 The global handle server

5.2 Local handle servers

5.3 Storing handles

[6.](#) Resolution of Handles

6.1 Clients

6.2 Resolution without caches

6.3 Resolution with caching

[7.](#) Administration

7.1 Tools for administration of handles

7.2 Firewalls

[8.](#) The Handle System Project

[9.](#) References

[10.](#) Authors' Addresses

1. Introduction

1.1 Overview

The Handle System provides identifiers for digital objects and other resources in distributed computer systems on networks, especially the Internet. These identifiers are known as handles. The system ensures that handles are unique and that they can be retained over long time periods.

Fundamentally, a handle is an identifier that has associated with it one or more fields of typed data. To resolve a handle is to present a handle to the system and have returned some or all of the associated data. In some applications, this data can be used to locate the item, but the system places no restriction on the types of data that can be stored with the handle. Handle servers are distributed computer systems that store handle data and provide a rapid service to resolve handles.

Implementation of the handle system by the Corporation for National Research Initiatives (CNRI) began in mid-1994 and is scheduled for completion by the end of 1995. This paper is an overview of the entire system.

1.2 Historical background

The design and implementation of the handle system has been part of the Computer Science Technical Reports project, funded by ARPA. One part of this project was to develop an architecture for the underlying infrastructure of the digital library. This defines digital objects with repositories to store them, and describes how handles are used to identify the digital objects. The architecture is described in a paper by Robert Kahn and Robert Wilensky [cnri.dlib/tn95-01]. The Corporation for National Research Initiatives (CNRI) and others are implementing Kahn/Wilensky repositories using the handle system.

During 1994/95, a team led by David Ely at CNRI implemented a global handle server, client libraries, and simple management tools. An early application, which is nearing completion, is its use in the deposit of digital objects at the Copyright Office of the Library of Congress.

The CNRI handle system is completely general in the types of network-accessible items that can be identified, where they are stored, and how they are accessed. It is not restricted to digital objects stored in repositories. Typical examples of other items that can be identified with handles include World Wide Web pages, e-mail addresses, or public keys.

The global handle system has been in operation since mid-[1994](#). During the first six months of 1995, CNRI has added a caching handle server for greater performance, advanced tools for administration of handles and naming authorities, and a proxy server for use with World Wide Web clients. During the second half of 1995, local handle servers are being added to provide a balance between local autonomy in managing handles and global, long-term availability of handles. Local handle servers will also allow excellent performance in resolving very large numbers of handles.

[1.3](#) The handle system and the Internet Engineering Task Force

While the handle server has been under development, the Internet Engineering Task Force (IETF) has been seeking a system for naming Internet objects. The term used by the IETF is Universal Resource Name (URN). [RFC 1737](#) gives a list of requirements. While the IETF has not yet completed specification of the syntax of URN, the handle system provides all the functions listed in [RFC 1737](#).

[2.](#) The handle system

[2.1](#) System components

The handle system contains the following parts.

- o Naming authorities are entities authorized to create new handles and store them, with their associated handle records, in handle servers.
- o Handle generators create new handles on behalf of naming authorities.

- o Handle servers store handles and provide a service to resolve them. There is a single global handle server and many associated local handle servers.
- o Client software is used for user applications to communicate with handle servers.
- o Caching servers are used to provide fast resolution of handles for clients and to minimize the frequency with which client software accesses other handle servers.
- o Proxy servers permit Web browsers and other clients to resolve handles.
- o Administrative tools are provided to create naming authorities, to create, modify, and delete handle records, and to create and maintain administrative groups.

Each handle server is implemented as a distributed computer system, which may have many server computers.

2.2 Policies and procedures

The long-term value of the handle system depends upon many independent sub-systems, operated by independent organizations and a good balance between local and global servers. The global handle server and well-managed local handle servers can ensure their own integrity, but can not guarantee that other local handle servers are well behaved, maintain acceptable levels of performance, or even continue to exist.

The following are planned to maintain the integrity of the system, with the minimal policies and procedures.

- o The global handle server is designed to be a high performance, highly available system.
- o Whenever a handle is stored on any handle server, a check for uniqueness is made. In particular, the global handle server ensures uniqueness amongst handles stored at the global level.
- o It is strongly recommended that all handles should be stored on the global handle server where long term persistence is required or the handle is used to identify valuable or otherwise important items.

3. Basic Concepts

3.1 Handles and handle records

A handle provides a name for an item. A handle record, as stored in a handle server, contains:

- the handle
- one or more fields of typed data
- administrative information

Here are some examples:

- o The item is a digital object stored in one or more repositories, as described by Kahn and Wilensky. The data field contains the identity of a repository or a set of repositories.

- o The item is an html page, stored on several World Wide Web servers. Each data field contains a URL.

Two important types of resource that are identified by handles are naming authorities and handle servers.

3.2 Resolution of handles

Resolution of handles is carried out by handle servers, at the request of a client. To resolve a handle, a handle server receives as input a handle and returns some or all of the fields of typed data in the corresponding handle record.

The client can request that all data fields in the handle record be returned or only those fields that contain data of a given type.

The preferred method of transport is by UDP, but some firewalls do not pass UDP packets. Therefore, TCP is provided as an option.

3.3 Indirect handle

An indirect handle is a special type of data field that can be held in a handle record. The data field contains the address of a handle server, with a specific data type to indicate that this is an indirect handle. A handle server addresses is either an IP address or a domain name.

One use of an indirect handle is to allow reorganization of handles amongst handle servers. Indirect handles are left as forwarding addresses. A second important use is described in [Section 5.2](#).

3.4 Syntax

A handle has the form:

n/d

Where n is a naming authority and d is an arbitrary string. The string d is unique, for that naming authority. The following are examples of valid handles:

berkeley.cs/1994.12.05.23.42.12;7
cnri.dlib.papers/tn95-137

The precise character sets allowed in n and d are still under discussion. All alphanumeric and certain other ASCII characters are allowed. Text is case insensitive. Within n, a period (.) has a special significance.

For identification in Internet applications, a handle may be preceded by "hdl:", "hdl://", "x-hdl:", or "x-hdl://" as in the following example:

hdl:cnri.dlib/tn95-01

4. Naming Authorities

4.1 The naming authority hierarchy

The name of a naming authority, *n*, consists of one or more strings, separated by periods. Examples are:

```
berkeley.cs  
cnri.cs-tr.technology
```

The high-order part of the name ("berkeley" in the first example) is issued by the global naming authority.

Example. The global naming authority issues the name "cnri". Future naming authorities, created by cnri might be "cnri.cs-tr" or "cnri.xiwt".

4.2 Creating naming authorities

Each naming authority, *n*, has at least one super-administrator who has full privileges for that naming authority, including permission to create a lower order naming authority, *n.n'*, with its own super-administrator. This super-administrator creates permissions for administration of handles within that naming authority, and can also create new naming authorities, *n.n'.n''*, and so on. Super-administrators can delegate privileges to other administrators, including the privilege of creating naming authorities.

Example. The super-administrator for "cnri.cs-tr" can create a naming authority "cnri.cs-tr.technology".

4.3 Primary handle server for a naming authority

Every naming authority has associated with it a primary handle server, denoted by *P*. When a new naming authority is created, the primary handle server is initially set to be the global handle server, *G*. Thereafter the administrator of the naming authority can designate any handle server as its primary handle server, *P*.

Whenever the naming authority, n , creates a handle, n/d , either the handle, n/d , is stored in P or an indirect handle is stored in P , indicating that n/d exists and pointing to a handle server that holds n/d . Thus the primary handle server of any naming authority has handle records for all naming authorities that the naming authority has created.

4.4 Handles for naming authorities

When a new naming authority, $n.n'$, is created, it is given a handle. The form of the handle is: $n.n'$. The data part is null. The data field of the handle record contains the address of the primary handle server, P .

The handle for the naming authority is stored both in the global handle server and in the primary handle server of n . Thus the global handle server has records for all naming authorities.

5. Handle Servers

5.1 The global handle server

The global handle server is a distributed system that stores and resolves handles. It is publicly accessible. The system is highly secure, is fault tolerant, and designed to run continuously. The global handle server is denoted by G .

One function of G is to store handle records for items that must be retained over very long periods of time, such as copyright registration, or legal records. G also stores handles for all naming authorities and local handle servers.

G is a public service and any client can ask G to resolve any handle. Since the handles for all naming authorities and registered handle servers are stored in G , and G is public, the name of every naming authority, n , and its primary handle server, P , are public and available to all clients.

5.2 Local handle servers

Local handle servers are a local option. They work in conjunction with the global handle server to store and resolve handles locally. They provide increased local control of handles, distribute the computing load between central and locally supplied equipment, and provide simple access controls to handle data. By storing individual handles on both a local and the global handle server, they can be used to back up each other.

Local handle servers can be created and operated by naming authorities or repositories. Other organizations, such as service bureaus, can also create and run local handle servers. For a local handle server to become a registered part of the overall handle system, it must be given a handle (by some naming authority). This handle is then stored in G, the global handle server.

Local handle servers are not public services. Permissions for a client to use local handle server to resolve a handle are set by the system administrators. Currently, the only such method of access control is by the IP address of the client that makes a query to the handle server.

Each local handle server is implemented as a set of one or more server computers. When several computers are used, handles are distributed amongst them using a hash table. For reasons of performance and reliability, data may be replicated across these computers, but this is hidden from client programs.

5.3 Storing handles

Each naming authority, *n*, has at least one super-administrator who has full privileges to create handles for that naming authority, and to attach administrative permissions to each handle. The administrative permissions for each handle include the right to modify or delete the handle or some of its handle data. These permissions are stored in the primary handle server.

Naming authorities can choose to store the handles that they create on any handle servers for which they have access permissions, local or global. When a handle is stored in several servers, one is declared to be authoritative. This can be the global handle server, G, the primary handle server, P, or another local handle server, subject to the naming authority having administrative permission to store handles on that handle server.

G is publicly accessible for handle resolution. If a handle is stored in G, then any client can resolve it. Handles stored on other handles servers can be resolved only by clients that have suitable permissions.

Example. The naming authority "cmu.cs.robotics" might choose G as authoritative for the handle to an important object, and also enter the handle in its primary handle server, P, for local use.

When n creates a handle, it makes a record in P, the primary handle server of naming authority n, with an indirect handle to each handle server that is able to resolve this handle. This indirect handle indicates that the handle exists and can be resolved by a query to the appropriate handle server. Access controls on P should be set so that any client with permission to query the handle server is able to read this indirect handle.

Example. The naming authority "cnri.cs-tr.technology" creates a handle "cnri.cs-tr.technology/d1" which is stored in the global handle server. An indirect handle is stored in the primary handle server indicating that a handle "cnri.cs-tr.technology/d1" is stored in the global handle server.

6. Resolution of Handles

6.1 Clients

The handle system provides a client library of routines, currently written in the C programming language. They interface with the global and local handle servers and with caching servers. They can be included in client programs that wish to contact handle servers. The interface specification for the client library will be documented and made public.

Caches are used by clients to reduce the load on the other handle servers, particularly the global handle server, G. Resolution of handles using caches is, in general, faster than resolution without caches.

The caching server is a shared cache to be used by a group of clients. The architecture also allows a cache to be incorporated within an individual client.

The recommended configuration is for any client, C, to have an assigned cache, C1. This can be integrated into the client or can be caching server shared by several clients. C1, itself, may be connected to a higher order caching server, C2. To avoid resolution involving many steps, the recommended configuration is to have no more than two levels of caches, C1 and C2.

A proxy server has been developed that acts as a client to the handle system for use with World Wide Web browsers and other clients. The client passes a handle to the proxy server which attempts to resolve it. If the handle can be resolved into one or more URLs, a URL is returned to the client.

The proxy server is configured as a separate server to be used by a group of clients. The recommended configuration is that every organization that wishes to use the handle system should provide both a caching and proxy server for its community.

The proxy server has been developed by CNRI in consultation with the National Center for Supercomputing Applications, the developer of Mosaic, but is intended to work with other clients that support proxies. Mosaic will be configured to use the proxy when a handle is specified in place of a URL. The proxy server will be supported by future releases of Mosaic. It is also compatible with the earlier proxy server developed by CERN and will be kept compatible with other proxy servers as they are developed by the World Wide Web community.

The rest of this section is an informal description of how a client, C, resolves any handle, h. A detailed description of the resolution algorithm is in preparation. Important details include: (a) each handle server can be implemented as one or more server computers; (b) checks are required to prevent looping through indirect handles; (c) the client may not have access permissions for all local handle servers; (d) the client request may ask for all the data in a handle record or data of specified types only; (e) because the local handle servers are independently managed, the client may encounter inconsistent data or unacceptably poor response from a server.

6.2 Resolution without caches

If the client, C, is not attached to any caching server, it uses the following steps to resolve a handle, h.

1. C sends a query to G.

If the handle record for h is stored in G, G resolves h.

Otherwise, G returns the address of P, the primary handle server of naming authority, n.

2. If h is not yet resolved, C sends h to P.

If h is stored in P and C has the correct access permissions, P resolves h.

Otherwise, if there is an indirect handle to another handle server, M, which stores h, P sends the client the address of M.

3. If h is not yet resolved, the client, C, sends h to M.

If the client has the correct access permissions, M resolves h. (If C does not have permission, it should try other handle servers that hold the handle.)

6.3 Resolution with caching

If the client, C, is connected to a cache, C1, resolution of h follows these steps:

1. The client, C, asks C1 to resolve h.

If the handle record of h is in the cache, the handle record is returned to C.

2. Otherwise, if the identity of P, the primary handle server of naming authority n, is stored in C1, C1 resolves the handle following steps 2 and 3 above in the description of resolution without caching.

3. If the handle has not been resolved, and C1 is connected to a higher cache, C2, C1 asks C2 to resolve both h and P, and pass the results to C1 to be saved in C1's cache.

If h and P are not in C2's cache, the request is passed to the next higher cache, until the handle is resolved until the highest cache is reached. (The recommended configuration is to have no more than two levels of cache.)

4. If there is no higher cache, then the cache sends a request to G asking for the resolution of h and P. The resolution algorithm then continues as in the description of resolution without caching..

7. Administration

7.1 Tools for administration of handles

Administrative data is stored in each handle record as a special data type. Access to this data is governed by access permissions specified for each handle separately.

Administrative tools are provided for creation of naming authorities; for creating, modifying, and deleting handles; for changing access permissions by individual or by group.

Two sets of tools are currently provided. The first uses electronic mail. The only security is to check the "from" field in the e-mail header. The second uses Mosaic forms. Security is by ID and password. A third level of security is under development. It is intended to use public key encryption.

7.2 Firewalls

The handle system is based on the UDP protocol. This enables a large number of transactions to be handled efficiently, but some security firewalls reject UDP packets. Therefore, the choice of UDP or TCP is provided as alternatives for the local handle server, caching server, and client library, but not for the global handle server.

8. The Handle System Project

During fall 1994, CNRI made the Phase 0 handle system available on the Internet. This included a single-computer global handle server, administration by e-mail, the basic client library and an X-Mosaic client. Phase 1 will be released in July 1995. It includes the distributed global handle server, caching handle server, proxy handle server, administration by e-mail or Mosaic forms, full client, and TCP support for firewalls. Phase 2 is scheduled for release later in 1995. It includes everything in Phase 1, plus the local handle server. The release date for security based on public key encryption is not yet scheduled.

The following have contributed to the handle system design and implementation: David Ely (project head), William Arms, Navjeet Chabbewal, Judith Grass, Timothy Kendall, Charles Orth, Ed Overly, John Stewart. We want to acknowledge the contribution on Robert Kahn, Robert Wilensky, and the other members of the Computer Science Technical Reports project.

This research was supported in part by the Advanced Research Projects Agency under Grant No. MDA-972-92-J-1029. Its content does not necessarily reflect the position or policy of the Government or CNRI, and no official endorsement should be inferred.

9. References

hdl:cnri.dlib/tn95-01 Kahn, Robert and Wilensky, Robert. "A framework for distributed digital object services". May, 1995.
(<http://www.CNRI.Reston.VA.US/home/cstr/arch/k-w.html>)

10. Authors' Addresses

William Arms (warms@cnri.reston.va.us)
David Ely (dely@cnri.reston.va.us)

Corporation for National Research Initiatives
Suite 100, 1895 Preston White Drive
Reston, VA 22091
Tel: (703) 620-8990

[Page 17]