

Directory Assistance Service

Status of this Memo

This document defines a mechanism by which a user-interface may access a textual DAP-like interface over a TCP/IP connection. This is a local mechanism. This memo provides information for the Internet community. It does not specify any standard. Distribution of this memo is unlimited.

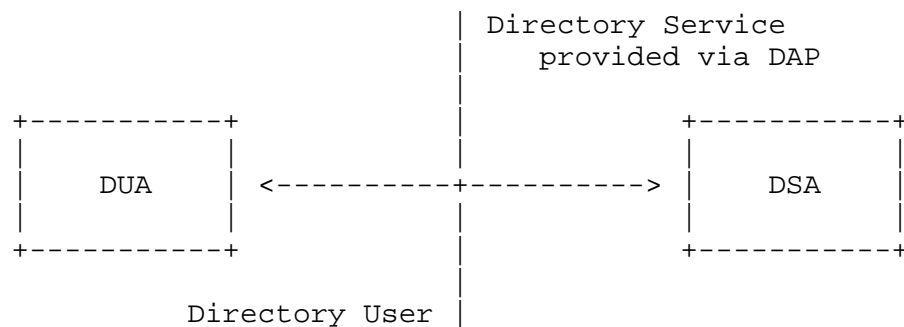
Table of Contents

| | |
|---------------------------------|----|
| 1. Introduction | 1 |
| 1.1 An Aside | 3 |
| 2. Protocol | 3 |
| 2.1 Control Connection | 4 |
| 2.1.1 Initialization | 4 |
| 2.1.2 Transactions | 4 |
| 2.1.2.1 INTR command | 4 |
| 2.1.2.2 STAT command | 5 |
| 2.1.2.3 QUIT command | 5 |
| 2.2 Data Connection | 5 |
| 2.2.1 Transactions | 5 |
| 2.2.2 Responses | 6 |
| 2.2.2.1 Numeric Responses | 6 |
| 2.2.2.2 'm' Response | 6 |
| 2.2.2.3 'y' Response | 6 |
| 2.2.2.4 'p' Response | 7 |
| 2.2.2.5 'e' Response | 7 |
| 2.2.2.6 'l' Response | 7 |
| 2.2.2.7 'd' Response | 8 |
| 2.2.2.8 'P' Response | 8 |
| 3. Example Interaction | 9 |
| 4. References | 10 |
| 5. Security Considerations..... | 11 |
| 6. Author's Address..... | 11 |

1. Introduction

The OSI Directory [1] provides a powerful infrastructure for the retrieval of information objects. This infrastructure can be used to support, e.g., white pages applications, application entity lookup, and so on.

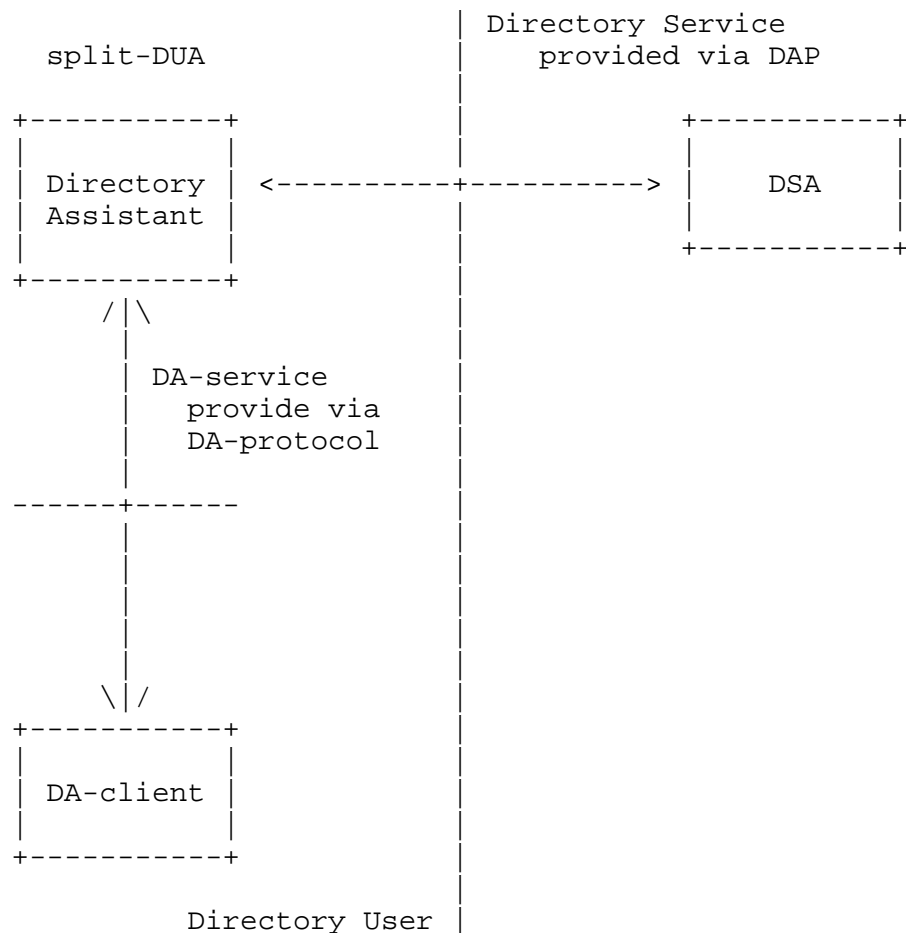
The Directory service is provided to applications through the Directory Access Protocol (DAP), which binds a Directory User Agent (DUA) to a Directory System Agent (DSA).



The DAP is an OSI application layer protocol which uses the rich OSI upper-layer infrastructure. Unfortunately, the coding investment to implement the DAP is significant. As such, it is difficult to host applications using the Directory on smaller workstations and personal computers.

This memo details a local mechanism which has been successfully used to separate the functionality of the DAP from the complexity of implementing the DAP. That is, a split-DUA model is used: the DAP is implemented on an entity (the "Directory Assistant"), which resides on a capable workstation or mainframe and exports a simpler interface, the "Directory Assistance" (DA) protocol, to other end-systems where the user-interface resides, termed the DA-client.

Since this mechanism provides assistance to applications wishing to access the Directory, it is termed the "Directory Assistance" (DA) service:



1.1. An Aside

This memo documents an already existing protocol, which was originally used to provide a split-DUA model within the same host. In the absence of detailed historical and implementational understanding, some of the mechanisms described may not appear intuitive.

2. Protocol

The DA service operates using two TCP connections: a control connection, and a data connection. The control connection defines the lifetime of an instance of the DA service; throughout this lifetime, several data connections may be established. However, at any given instant, between zero and one data connections will be in progress.

The DA service is provided by the "Directory Assistant", which consists of two entities: a DA-server, which manages the control connection; and, a DAP-listener, which responds to commands on the data connection. The DA-server oversees the behavior of the DAP-listener.

2.1. Control Connection

Data sent over the control connection consists of a series of transactions. NVT-ASCII is used to express these transactions. Each transaction consists of the client sending a directive--a line of text terminated by CR-LF; the DA-server returns a response--a line of text terminated by CR-LF. All responses from a DA-server start with either "+OK" or "-ERR" depending on whether the transaction was successful.

2.1.1. Initialization

A DA-server listens on TCP port 411 for incoming connections. Upon establishing a control connection, the DA-server returns a response indicating whether the service has been started. If successful, the response contains an IP-address and a TCP port, expressed in NVT-ASCII, and separated by one or more instances of the space character. This information corresponds to the TCP-endpoint that the DAP-listener will use for the data connection.

Note that the DA-server and DAP-listener need not reside at the same IP-address. In the future, DA-servers may employ a internal protocol for load-balancing purposes.

If the DA service can not be started, an error response is returned and the control connection is closed.

2.1.2. Transactions

All transactions with the DA-server consist of a command followed by zero or more arguments, separated by the space character.

2.1.2.1. INTR command

The INTR command takes no arguments.

The INTR command is used to interrupt any DAP transaction currently in progress.

The INTR command always returns success.

2.1.2.2. STAT command

The STAT command takes no arguments.

The STAT command is used to verify that the DAP-listener is available.

The STAT command returns success only if the DAP-listener is still active.

2.1.2.3. QUIT command

The QUIT command takes no arguments.

The QUIT command is used to terminate the DA service.

The QUIT command always returns success.

2.2. Data Connection

Data sent over a data connection consists of a single DAP-transaction. NVT-ASCII is used to express these transactions. Each transaction consists of the client sending a command--a line of text terminated by the LF-character; the DAP-listener returns zero or more responses, each with a specific termination sequence. All responses from a DAP-listener start with a single identifying character. If the character is a digit (0-9), then the termination sequence consists of a closing the data connection; otherwise, if the character is a lower-case letter (a-z), then the response is interactive and is terminated by the LF-character.

2.2.1. Transactions

All transactions with the DAP-listener consist of a command followed by zero or more arguments, separated by the space character. Double-quotes may be used to prevent separation of tokens.

The command set is taken from the DISH program:

| | |
|------------|---------------------------|
| add | add a new entry |
| bind | connect to the Directory |
| compare | compare entry's attribute |
| delete | delete an entry |
| fred | back-end to FrED |
| list | list children |
| modify | modify an existing entry |
| modifyrdrn | modify an entry's name |
| moveto | move to a position |

| | |
|-----------|-------------------------------|
| search | search for an object |
| showentry | show an entry |
| showname | show an entry's name |
| squid | status of dish |
| unbind | disconnect from the Directory |

See [2] for a complete list of commands and arguments.

Note that commands and arguments are in lower-case, and may abbreviated to any unique prefix.

2.2.2. Responses

There are two kinds of responses: numeric-responses, which consist of arbitrary text; and, letter-responses, which consist of brief text, and expect further interaction from the client.

2.2.2.1. Numeric Responses

If the response is '1', then the DAP-transaction terminated normally; if the response is '2', then the DAP-transaction failed; if the response is '3', then the DAP-transaction was a search returning more than one result and one of the -hitone or -list option was selected for the search; if the response is '4', then the DAP-transaction terminated normally and the remainder of this line consists of the name of an entry (see the 'd' Response below); if the response is '5', then all children of an entry were found by the DAP-transaction. Once the response is completely sent, the DAP-listener closes the data connection.

Note that although numeric responses utilize ASCII, they are not NVT-ASCII; in particular, the LF-character is used to indicate end-of-line, rather than the CR-LF line termination sequence of NVT-ASCII.

2.2.2.2. 'm' Response

The 'm' response contains a one-line message which should be presented to the user.

At this point, the client returns a response consisting of 'm' followed by the LF-character. The client should then continue reading from the existing data connection.

2.2.2.3. 'y' Response

The 'y' response contains a yes/no question which should be presented to the user. After querying the user, the response (either 'y' or

'n'), followed by the LF-character, should be sent to the DAP-listener. The client should then continue reading from the existing data connection.

2.2.2.4. 'p' Response

The 'p' response contains a password-prompt which should be presented to the user. After querying the user, the client returns a response consisting of 'p' followed by the password supplied by the user followed by the LF-character. The client should then continue reading from the existing data connection.

2.2.2.5. 'e' Response

The 'e' response is used to ask the user to edit some text. Following the 'e' character is a decimal number in ASCII followed by the LF-character, indicating the number of octets that should be presented to the user for editing (these octets may include LF-characters).

At this point, the client returns a response consisting of a single character followed by a LF-character. If the character is 'e', the edit is aborted (e.g., the text is too large), and the client should then continue reading from the existing data connection.

Otherwise, the DAP-listener sends the indicated number of octets corresponding to the buffer that the user is to edit. After the user edits the buffer, one of two responses should be sent.

If the user aborted the edit, the response sent to the DAP-listener is a single character 'e', followed by the LF-character.

Otherwise, the response consists of any single character other than indicating the number of octets immediately following that resulted from the user-edit.

Regardless of the outcome, the client should then continue reading from the existing data connection.

2.2.2.6. 'l' Response

The 'l' response contains an entry for a selection list to be presented to the user. The form of this entry consists of two strings separated by the '\$' character, and terminated by the LF-character. The first string is a user-friendly name, suitable for display to the user; the second string is a fully-qualified Distinguished Name in textual format.

At this point, the client returns a response consisting of 'l' followed by the LF-character.

The client should continue to accumulate selection entries until an LF-character.

At this point, the user should be asked to select one or more of the selection entries. After this selection, the client sends back a response consisting of 'L' followed by one or more decimal numbers in ASCII followed by the LF-character. The numbers are separated by spaces, and correspond to the entries selected by the user. (The entry corresponding to the first 'l' response is numbered 1, etc.)

The client should then continue reading from the existing data connection.

2.2.2.7. 'd' Response

The 'd' response contains a name that the client may be interested in. The form of this name consists of two strings separated by the '\$' character, and terminated by the LF-character. The first string is a user-friendly name, suitable for display to the user; the second string is a fully-qualified Distinguished Name in textual format.

At this point, the client returns a response consisting of 'd' followed by the LF-character. The client should then continue reading from the existing data connection.

2.2.2.8. 'P' Response

The 'P' response is used to transmit a picture to the client. Following the 'P' character is a decimal number in ASCII followed by a name and then the LF-character. The decimal number indicates the size of the picture. The name contains three strings separated by the '\$' character. The first string is the name of the attribute corresponding to the picture, in textual format; the second string is a user-friendly name, suitable for display to the user; and, the third string is a fully-qualified DistinguishedName in textual format.

At this point, the client returns a response consisting of a single character followed by a LF-character. If the character is 'P', the picture will not be sent (e.g., the image is too large), and the client should then continue reading from the existing data connection.

Otherwise, the DAP-listener sends the indicated number of octets corresponding to the picture. The picture is encoded using the PBM

format from the PBMPLUS package.

Regardless of the outcome, the client should then continue reading from the existing data connection.

3. Example Interaction

In the text that follows, "S:" refers to the DA-server, "L:" refers to the DAP-listener, "C:" refers to the client talking to the DA-server, and, "I:" refers to the client talking to the DAP-listener.

```
S: <wait for connection on TCP port 411>

C: <open connection to DA-server>
L: <wait for connections>
S: +OK 192.33.4.21 32867

I: <open connection to DAP-listener>
I: bind -simple -user "@c=US@cn=Manager"
L: pc=US@cn=Manager
-- client asks user for password for "c=US@cn=Manager"
I: psecret
L: <closes connection, signaling success but no response>

-- since response was null, client verifies that DAP-listener
-- is still operating...
C: STAT
S: +OK

I: <open connection to DAP-listener>
I: fred -expand "@"
L: 5
    North America$l=North America
    US$c=US
    ...
L: <closes connection>

I: <open connection to DAP-listener>
I: fred -ufn rose,psi,us
L: 1
    <followed by much data>
L: <closes connection>

I: <open connection to DAP-listener>
I: fred -ufn -list,rose,ps,us
L: lHewlett-Packard, US$c=US@o=Hewlett-Packard
I: l
L: lPerformance Systems International, US$c=US@o=Performance...
```

```
I: 1
L: 1Rutgers University, US$c=US@o=Rutgers University
I: 1
L: Lps
-- client presents selection list to user asking to select
-- matches for 'ps', user selects the 2nd
I: L 2
L: dManager, US$c=US@cn=Manager
I: d
L: 4Marshall Rose, ...$c=US@o=Performance...
    <followed by much data>
L: <closes connection>

I: <open connection to DAP-listener>
I: fred -ufn -list,schoffstall,ps,us
L: 33 matches found.
Martin Schoffstall, ...$c=US@o=Performance...
Marvin Schoffstall, ...$c=US@o=Performance...
Steve Schoffstall, ...$c=US@o=Performance...
L: <closes connection>

C: QUIT
L: <stop listening for connections>
S: +OK
C: <close connection>

S: <wait for next connection>
```

4. References

- [1] Information Processing - Open Systems Interconnection - The Directory, International Organization for Standardization. International Standard 9594, (1988).
- [2] Kille, S., Robbins, C., Roe, M., and A. Turland, "The ISO Development Environment: User's Manual", Volume 5: QUIPU, Performance Systems International, January 1990.

5. Security Considerations

Security considerations are not discussed in this memo.

6. Author's Address

Marshall T. Rose
PSI, Inc.
PSI California Office
P.O. Box 391776
Mountain View, CA 94039

Phone: (415) 961-3380

EMail: mrose@psi.com