Open Software Foundation **R. Salz (OSF)**
Request For Comments: 81.1
July 1995

DCE ASSIGNED VALUES

1. INTRODUCTION

DCE includes a number of architected values in a variety of areas. The official documentation for these values is the ''Application Environment Specification'' (AES) volumes. Because the AES production cycle is long, this document can be considered to be an update, reflecting changes that are likely to appear in the ''next edition'' of the appropriate AES.

Note that until the AES is published, OSF does not officially condone or endorse any of the values enumerated here. In particular, we take no position on conflicting assignments. We do, however, encourage licensees and implementors to take interoperability as their foremost concern and work together when at all possible.

Along those lines, we are willing to register and define almost anything. In addition to the areas listed below, we are willing to ''open up'' other parts of DCE by adding new RPC PDU types, additional threads API's, and so on.

In general, we can work in two ways. First, we can act as a semi-formal registry by issuing updates to this document. In this case requestors will need to provide descriptions of the semantics associated with the value, so that other implementors can provide similar functionality and/or maintain interoperability. Second, we can ''mark off'' areas and guarantee that OSF DCE will never use certain values. In this case implementors are free to use any such values, subject to the risk of ''bumping into'' another implementor's use of the same value. (Values in this group can be useful for prototyping or dedicated enterprise installations.)

Request should be made by sending email to `dce-registry@osf.org` where they will be handled on a case-by-case basis.

Updates to this document will be issued as needed. To get the latest draft version, send email to `dce-registry@osf.org`.

1.1. Changes Since Last Publication

Besides some minor reformatting, the substantive changes since the previous version of this document ([RFC 81.0]) are the following:

(a) Defined the `ncacn_osi_mosi`, `ncadg_osi_clns`, `ncacn_vns_spp`, and `ncadg_vns_ipc`, protocol sequences.

(b) Defined the `snm`, `mcl`, `ems`, and `dcf` components.

(c) Defined ISV message blocks 100 and 101.

2. CORE DCE CHANGES

This section documents new values defined for DCE.

2.1. Protocol Sequences

Seven new protocol sequences are defined. This updates Appendices B and I of [RPC AES]. Also included with each protocol sequence is its corresponding protocol tower identifier (name and value).

(a) `ncacn_at_dsp` —
Connection-oriented, Appletalk NBP-style addresses, Appletalk's data stream protocol.
Protocol tower ID: `at` (0x18).

(b) `ncadg_at_ddp` —
Connectionless, Appletalk NBP-style addresses, Appletalk's datagram delivery protocol.

Protocol tower ID: `dsp` (0x16).

(c) `ncadg_nb` —
Connectionless, NetBios (over all available protocols).
Protocol tower ID: `ddp` (0x17).

(d) `ncacn_vns_spp` —
Connection-oriented, VINES StreetTalk addresses, VINES SPP protocol.
Protocol tower ID: `nb` (0x19).

(e) `ncadg_vns_ipc` —
Connectionless, VINES StreetTalk addresses, VINES IPC protocol.
Protocol tower ID: `spp` (0x1a).

(f) `ncacn_osi_mosi` —
Connection-oriented, over a seven-layer minimal OSI (''mOSI'') stack.
Protocol tower ID: `ipc` (0x1b).

(g) `ncadg_osi_clsn` —
Connectionless, over a seven-layer OSI stack.
Protocol tower ID: `vns` (0x1c).

[In the following paragraphs, ''left'' and ''right'' refer to the technical terms for the ''sides'' of a protocol tower floor.]

The `dsp` or `ddp` protocol occupies the third floor of a DCE RPC tower. The left side of the floor is the protocol identifier (0x16 or 0x17) and the right side is a counted ASCII string naming the endpoint (or object). The `at` protocol occupies the fourth floor of a DCE RPC tower. The left side of the floor is the protocol identifier (0x18) and the right side is a counted ASCII string containing an NBP-style name in the form `name@zone`; the zone named `*` is used to indicate the local zone if no zone is specified.

These two floors are converted to an Appletalk name by concatenating the following elements: `DceDspRpc` or `DceDdpRpc`, a space, the `endpoint`, a colon, and the `name@zone`. The well-known endpoint (or object name) of the endpoint mapper is `Endpoint Mapper`.

The `spp` or `ipc` protocol occupies the third floor of a DCE RPC tower. The left side of the floor is the protocol identifier (0x1A or 0x1B) and the right side is a two-byte port number (high-byte first) representing the endpoint.

The `vns` protocol occupies the fourth floor of a DCE RPC tower. The left side of the floor is the protocol identifier (0x1C) and the right side is a counted ASCII string containing a StreetTalk name. The name is of the form `Item@Group@Organization` where the `Item` can have no more then 31 characters and the other two components can have no more than 15 characters each. The well-known endpoint of the endpoint mapper for `ncacn_vns_spp` or `ncadg_vns_ipc` is 385.

For details on the protocol identifiers and tower floor contents for the `ncacn_osi_mosi` and `ncadg_osi_clsn` protocol sequences, see ISO/IEC ISP 11188-3-mOSI (forthcoming) and ISO/IEC ISP 11188-4-CLS, respectively.

2.2. CDS Attribute Names

A new attribute is defined and used by the DCE 1.1 release. This updates Appendix J of [RPC AES].

(a) `RPC_Codesets` —
1.3.22.1.1.5
{iso(1) identified-org(3) osf(22) dce(1) rpc(1) RPC_Codesets(5)}

2.3. Other ISO OID's

For the sake of completeness, we document the following additional OID's also used or reserved by DCE.

(a) `RPC` —
1.3.22.1.1
{iso(1) identified-org(3) osf(22) dce(1) rpc(1)}

(b) `GDS` —
1.3.22.1.2
{iso(1) identified-org(3) osf(22) dce(1) gds(2)}

(c) `CDS` —
1.3.22.1.3
{iso(1) identified-org(3) osf(22) dce(1) cds(3)}

(d) `Security` —
1.3.22.1.5
{iso(1) identified-org(3) osf(22) dce(1) sec(5)}

(e) `SEC_Replica` —
1.3.22.1.5.1
{iso(1) identified-org(3) osf(22) dce(1) sec(5) SEC_Replica(1)}

(f) `Federated Naming` —
1.3.22.1.6
{iso(1) identified-org(3) osf(22) dce(1) xfn(6)}

2.4. New RPC Fault Codes

Two new fault codes are defined. This updates the table in section N.2 of Appendix N of [RPC AES]:

(a) `const long nca_s_fault_object_not_found = 0x1C000024;`

(b) `const long nca_s_fault_no_client_stub = 0x1C000025;`

A server sends the `nca_s_fault_object_not_found` code when the client has attempted to bind to an existing object (for example by CDS name) but the object was not known to the server. A server sends the `nca_s_fault_no_client_stub` code when a required client stub module is not linked into the executable (e.g., when a server stub or application code attempts to make an RPC).

2.5. Serviceability Components

The full list of the secure core 1.1 serviceability components in the DCE 1.1 release does not appear to be defined anywhere. They are as follows:

(a) `aud` (Auditing)

(b) `cds` (CDS)

(c) `cfg` (DCE Configuration)

(d) `csr` (Code Set Registry)

(e) `dcp` (`dcecp`)

(f) `dhd` (`dced`)

(g) `dts` (DTS)

(h) `gds` (GDS)

(i) `gss` (GSSAPI)

(j) `idl` (IDL compiler)

(k) `lib` (DCE utilities libraries)

(l) `rpc` (RPC)

(m) `sad` (Security administration tools)

(n) `sec` (Security runtime and server)

(o) `smp` (Sample code for DCE documentation)

(p) `svc` (Serviceability and messaging)

(q) `tcl` (TCL interpreter)

(r) `thd` (Threads)

(s) `uid` (`uuidgen`)

The DFS components are not listed.

2.5.1. New serviceability components

The following serviceability components are defined.

(a) `dcf` (Management configuration)

(b) `ems` (Event management system)

(c) `mcl` (Management object class library)

(d) `pkc` (Public-key certification)

(e) `snm` (SNMP subagent)

In addition, OSF guarantees that it will never define a serviceability component that starts with the two-letter sequence `qz`.

3. ISV STATUS CODE ASSIGNMENTS

DCE 1.1 includes a public API and mechanism for interoperable status codes and unique message identifiers [RFC 24.2]. In order to obtain a block of status codes, send email to `dce-registry@osf.org` including the product name, the vendor name, and the text of the first message. This text is used by `dce_error_inq_text` to identify the software issuing the status code.

In addition, OSF guarantees to never assign component codes less then 50.

The following codes are currently assigned. The first line specifies the message block and the first message in the block. The second line specifies the product and company name.

(a) 100 —
`"Distributor/Agent Extension"`
Distributor/Agent Extension, by Tandem Computers.

(b) 101 —
`"NonStop DCE"`
NonStop DCE, by Tandem Computers.

REFERENCES

[RPC AES] Open Software Foundation, *OSF DCE Application Environment Specification/Distributed Computing — Remote Procedure Call (RPC)*, November 10, 1993.

[RFC 24.2] R. Salz, DCE-RFC 24.2, ‘‘Making the DCE 1.1 Serviceability and Message API’s Public’’, April, 1993.

[RFC 81.0] R. Salz, OSF-RFC 81.0, ‘‘DCE Assigned Values’’, March 1995.

AUTHOR'S ADDRESS

Rich Salz
Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142
USA

Internet email: rsalz@osf.org
Telephone: +1-617-621-7253