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 81.1]) are the following:

(a) Defined the `ncacn_nb_stream`, `ncadg_nb_dgram`, `ncacn_unix_stream`, and `ncadg_unix_dgram` protocol sequences.

(b) Defined the `pks`, `ssa`, and `web` components.

(c) Defined the DCE management OIDs.

(d) Defined the audit event ids and event class ids.

(e) Defined the extended attributes for security and `dced`.

2. CORE DCE CHANGES

This section documents new values defined for DCE.

2.1. Protocol Sequences

Eleven new protocol sequences are defined. This updates Appendices B and I of [RPC AES].

(a) `ncacn_at_dsp`: Connection-oriented, Appletalk NBP-style addresses, Appletalk’s data stream protocol.

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

(c) `ncadg_nb`: Connectionless, NetBios (over all available protocols).

(d) `ncacn_vns_spp`: Connection-oriented, VINES StreetTalk addresses, VINES SPP protocol.

(e) `ncadg_vns_ipc`: Connectionless, VINES StreetTalk addresses, VINES IPC protocol.

(f) `ncacn_osi_mosi`: Connection-oriented, over a seven-layer minimal OSI (‘‘mOSI’’) stack.

(g) `ncadg_osi_clsn`: Connectionless, over a seven-layer OSI stack.

(h) `ncacn_nb_stream`: Connection-oriented, using NETBIOS datagram protocols.

(i) `ncadg_nb_dgram`: Connectionless, using NETBIOS session protocols.

(j) `ncacn_unix_stream`: Connection-oriented, using Unix Domain sockets.

(k) `ncadg_unix_dgram`: Connectionless, using Unix Domain sockets.

The corresponding protocol tower identifiers are as follows:

```
at      0x18
dsp     0x16
ddp     0x17
nb      0x19
spp     0x1A
ipc     0x1B
vns     0x1C
unix    0x20
null    0x21
netbios 0x22
```

[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.

The `unix` protocol occupies the fourth floor of a DCE RPC tower. The left side of the floor is the protocol identifier (0x20) and the right side is a ‘‘C’’ string: a zero-terminated array of ASCII bytes. The value is the name of the Unix-domain socket and must not exceed 108 bytes.

The `nb` protocol occupies the fifth floor of a DCE RPC tower. The left side of the floor is the protocol identifier (0x21) and the right side is the 16-byte NETBIOS name.

The `null` protocol can occupy any floor and is used as a ‘‘spacer’’ where needed. The left side is the protocol identifier (0x22); there is no right side. This is used as the fifth floor of the IBM NETBIOS protocol sequences.

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].

```
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.

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

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

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

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

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

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

DCE SNMP            1.3.22.1.7
                    {iso(1) identified-org(3)
                    osf(22) dce(1) snmp(7)}

DCE MIB             1.3.22.1.7.1
                    {iso(1) identified-org(3)
                    osf(22) dce(1) snmp(7) mib(1)}

DCE Subagent        1.3.22.1.7.2
                    {iso(1) identified-org(3)
                    osf(22) dce(1) snmp(7) subagent(2)}
```

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]:

```
const long nca_s_fault_object_not_found = 0x1C000024;
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:

```
aud    Auditing
cds    CDS
cfg    DCE Configuration
csr    Code Set Registry
dcp    dcecp
dhd    dced
dts    DTS
gds    GDS
gss    GSSAPI
idl    IDL compiler
lib    DCE utilities libraries
rpc    RPC
sad    Security administration tools
sec    Security runtime and server
smp    Sample code for DCE documentation
svc    Serviceability and messaging
tcl    TCL interpreter
thd    Threads
uid    uuidgen
```

The DFS components are not listed.

2.5.1. New serviceability components

The following serviceability components are defined.

```
dcf    Management configuration
dms    Distributed Measurement System
ems    Event management system
mcl    Management object class library
pkc    Public-key certification
pks    Private-key storage server
psm    Personal Security Module
ssa    SNMP SubAgent
web    DCE-Web Advanced Technology Offering
```

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

2.6. Audit Events

The DCE 1.1 audit facility [RFC 29] identifies an audit event by a 32-bit number, partitioned into an `event set-id` and an `event id`. For management purposes, events are collected into classes. An event class is also identified by a 32-bit number, partitioned into an `class set-id` and an `class id`. OSF will register event set-id's and class set-id's

Like Internet IP addresses, both events and event classes come in different formats, which determine how many bits are allocated for the set/id. Using a binary notation (MSB on the left), the formats are as follows:

```
Event Number Format A        0sss vvvv vvvv vvvv
Event Number Format B        10ss ssss vvvv vvvv
```

```
Event Number Format C         110s ssss ssss vvvv
Event Number Format D         1110 vvvv vvvv vvvv
Event Number Format E         1111 --reserved--

Event Class Number Format A   01ss ssss iiii iiii
Event Class Number Format B   10ss ssss ssss iiii
Event Class Number Format C   110i iiii iiii iiii
Event Class Number Format D   111- --reserved--

where
s  Indicates the event or event class set-id
v  Indicates the event id
i  Indicates the event class id
```

Values for Event Number Format D and Event Class Number Format C will never be assigned by OSF and can be freely used for inter-cell or development work.

2.7. OSF Audit Assignments

The following event numbers are used by OSF and its affiliates:

```
0x00vvvvvv  OSF (0x000001vv secd; 0x000002vv dts;
            0x000003vv audit; 0x000004vv dce-web)
0x01vvvvvv  X/Open
```

The following event class numbers are currently used by OSF and its affiliates:

```
0x0002  DTSD state modification
0x0003  DTSD state query
0x0004  DTSD time synchronization
0x0005  DTSD time provider interations
0x000A  SECD authentication/cryptographic events
0x000B  SECD state modification
0x000C  SECD controlled access events
0x000D  SECD object queries, lookups, or tests
0x000E  SECD configuration
0x0030  AUDITD filter insertion
0x0031  AUDITD filter query
0x0032  AUDITD state modification
0x0033  AUDITD state query
0x0040  DCE-Web security domain gateway
0x0100  XDAS Generic Audit Events
```

3. ERA-STYLE ATTRIBUTES

DCE 1.1 provides an extended attribute facility [RFC 6] that defines a schema management interface, and a set of datatypes that lets an application have an common extensible method of storing arbitrary data. Both the security service and the DCE host daemon [RFC 47] use this facility.

3.1. Security Service ERA's

The following UUID's and names are defined for the security service. For explanation of their semantics, consult the DCE documentation.

```
6c9d0ec8-dd2d-11cc-abdd-080009353559  pre_auth_req
    0=NONE, 1=PADATA_ENC_TIMESTAMPS, 2=PADATA_ENC_THIRD_PARTY
689843ce-dd2d-11cc-a3e1-080009353559  pwd_val_type
    0=NONE, 1=USER_SELECT, 2=USER_CAN_SELECT,
    3=GENERATION_REQUIRED
6a93b8f2-dd2d-11cc-9be7-080009353559  pwd_mgmt_binding
```

```
    Binding to server exporting the password management
    interfaces
c5949eba-384a-11cd-8cba-080009353559  X500_DN
    The principal's X500 Distinguished Name
c6a51456-384a-11cd-b6ef-080009353559  X500_DSA_Admin
    List of DSAs that the principal is allowed to administer
63005af0-dd2d-11cc-9be7-080009353559  disable_time_interval
    Number of seconds to disable account
657eb68c-dd2d-11cc-8990-080009353559  max_invalid_attempts
    Number of invalid attempts allowed before account is
    disabled
bc51691e-dd2d-11cc-9866-080009353559  passwd_override
    The ability to not be restricted by passwd expiration
6d8d97bc-dd2d-11cc-b1cc-080009353559  login_set
    The login set identifier
```

3.2. Host Daemon Attributes

The following attributes are defined for `dced`:

```
008b47dd-6ec9-1d6a-9ac7-0000c09ce054  hostdata/data
    The contents of a hostdata object as a set of strings
764fd860-3b6f-11cd-b254-08000925634b  hostdata/bindata
    The contents of a hostdata object as an array of bytes
b574524e-6b37-11cd-8ec2-08000925634b  srvrconf/dtsconfig
    DTSD configuration information
041f9efc-6b39-11cd-8848-08000925634b  srvrconf/
                                      additional_environ
    Additional environment strings to pass to the started
    server
```

4. 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.

```
100     "Distributor/Agent Extension"
        Distributor/Agent Extension, by Tandem Computers
101     "NonStop DCE"
        NonStop DCE, by Tandem Computers
```

5. ISV AUDIT ASSIGNMENTS

The following event numbers are currently assigned:

```
0x81eeeeee  Intraverse secure Internet technologies
            by DASCOM (CP Labs, Ltd.).
```

The following event classes are currently assigned:

```
0x800001cc  Intraverse audit events
            by DASCOM (CP Labs, Ltd.).
```

REFERENCES

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

[RFC 6] J. Pato, DCE-RFC 6.0 ''A Generic Interface for Extended Registry Attributes'', June, 1992.

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

[RFC 29.2] S. Luan, R. Weisz, ''Design of an Audit Subsystem for DCE — Functional Specification'', October, 1994.

[RFC 47] J. Bowe, D. Mackey, R. Salz, P. Wang, ''DCED: The DCE Host Daemon — Functional Specification'', April, 1994.

[RFC 81.1] R. Salz, OSF-RFC 81.1 ''DCE Assigned Values'', July, 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