
Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document describes the Open Network Computing (ONC) Remote Procedure Call (RPC) version 2 protocol as it is currently deployed and accepted. This document obsoletes RFC 1831.
Table of Contents

1. Introduction .................................................... 3
   1.1. Requirements Language ..................................... 3
2. Changes since RFC 1831 ........................................ 3
3. Terminology .................................................... 3
4. The RPC Model .................................................. 4
5. Transports and Semantics ....................................... 5
6. Binding and Rendezvous Independence .......................... 7
7. Authentication .................................................. 7
8. RPC Protocol Requirements ..................................... 7
   8.1. RPC Programs and Procedures .............................. 8
   8.2. Authentication, Integrity, and Privacy .................... 9
   8.3. Program Number Assignment ................................ 10
   8.4. Other Uses of the RPC Protocol ............................ 10
   8.4.1. Batching ............................................. 10
   8.4.2. Broadcast Remote Procedure Calls ...................... 11
9. The RPC Message Protocol ..................................... 11
10. Authentication Protocols ...................................... 15
11. Record Marking Standard ..................................... 16
12. The RPC Language ............................................. 16
   12.1. An Example Service Described in the RPC Language .... 17
   12.2. The RPC Language Specification .......................... 18
   12.3. Syntax Notes .......................................... 18
13. IANA Considerations ......................................... 19
   13.1. Numbering Requests to IANA ............................. 19
   13.2. Protecting Past Assignments ............................. 19
   13.3. RPC Number Assignment .................................. 19
   13.3.1. To be assigned by IANA .............................. 20
   13.3.2. Defined by Local Administrator ....................... 20
   13.3.3. Transient Block ..................................... 20
   13.3.4. Reserved Block ...................................... 21
   13.3.5. RPC Number Sub-Blocks ............................... 21
   13.4. RPC Authentication Flavor Number Assignment .......... 22
   13.4.1. Assignment Policy ................................... 22
   13.4.2. Auth Flavors vs. Pseudo-Flavors ..................... 23
13.5. Authentication Status Number Assignment ................ 23
   13.5.1. Assignment Policy ................................... 23
14. Security Considerations ...................................... 24
Appendix A: System Authentication ................................ 25
Appendix B: Requesting RPC-Related Numbers from IANA .......... 26
Appendix C: Current Number Assignments ........................ 27
Normative References ........................................... 62
Informative References ........................................... 62
1. Introduction

This document specifies version 2 of the message protocol used in ONC Remote Procedure Call (RPC). The message protocol is specified with the eXternal Data Representation (XDR) language [RFC4506]. This document assumes that the reader is familiar with XDR. It does not attempt to justify remote procedure call systems or describe their use. The paper by Birrell and Nelson [XRPC] is recommended as an excellent background for the remote procedure call concept.

1.1. Requirements Language

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 [RFC2119].

2. Changes since RFC 1831

This document obsoletes [RFC1831] as the authoritative document describing RPC, without introducing any over-the-wire protocol changes. The main changes from RFC 1831 are:

- Addition of an Appendix that describes how an implementor can request new RPC program numbers, authentication flavor numbers, and authentication status numbers from IANA, rather than from Sun Microsystems
- Addition of an "IANA Considerations" section that describes past number assignment policy and how IANA is intended to assign them in the future
- Clarification of the RPC Language Specification to match current usage
- Enhancement of the "Security Considerations" section to reflect experience with strong security flavors
- Specification of new authentication errors that are in common use in modern RPC implementations
- Updates for the latest IETF intellectual property statements

3. Terminology

This document discusses clients, calls, servers, replies, services, programs, procedures, and versions. Each remote procedure call has two sides: an active client side that makes the call to a server side, which sends back a reply. A network service is a collection of
one or more remote programs. A remote program implements one or more remote procedures; the procedures, their parameters, and results are documented in the specific program’s protocol specification. A server may support more than one version of a remote program in order to be compatible with changing protocols.

For example, a network file service may be composed of two programs. One program may deal with high-level applications such as file system access control and locking. The other may deal with low-level file input and output and have procedures like "read" and "write". A client of the network file service would call the procedures associated with the two programs of the service on behalf of the client.

The terms "client" and "server" only apply to a particular transaction; a particular hardware entity (host) or software entity (process or program) could operate in both roles at different times. For example, a program that supplies remote execution service could also be a client of a network file service.

4. The RPC Model

The ONC RPC protocol is based on the remote procedure call model, which is similar to the local procedure call model. In the local case, the caller places arguments to a procedure in some well-specified location (such as a register window). It then transfers control to the procedure, and eventually regains control. At that point, the results of the procedure are extracted from the well-specified location, and the caller continues execution.

The remote procedure call model is similar. One thread of control logically winds through two processes: the caller’s process and a server’s process. The caller first sends a call message to the server process and waits (blocks) for a reply message. The call message includes the procedure’s parameters, and the reply message includes the procedure’s results. Once the reply message is received, the results of the procedure are extracted, and the caller’s execution is resumed.

On the server side, a process is dormant awaiting the arrival of a call message. When one arrives, the server process extracts the procedure’s parameters, computes the results, sends a reply message, and then awaits the next call message.

In this model, only one of the two processes is active at any given time. However, this model is only given as an example. The ONC RPC protocol makes no restrictions on the concurrency model implemented, and others are possible. For example, an implementation may choose...
to have RPC calls be asynchronous so that the client may do useful work while waiting for the reply from the server. Another possibility is to have the server create a separate task to process an incoming call so that the original server can be free to receive other requests.

There are a few important ways in which remote procedure calls differ from local procedure calls.

- Error handling: failures of the remote server or network must be handled when using remote procedure calls.
- Global variables and side effects: since the server does not have access to the client’s address space, hidden arguments cannot be passed as global variables or returned as side effects.
- Performance: remote procedures usually operate at one or more orders of magnitude slower than local procedure calls.
- Authentication: since remote procedure calls can be transported over unsecured networks, authentication may be necessary. Authentication prevents one entity from masquerading as some other entity.

The conclusion is that even though there are tools to automatically generate client and server libraries for a given service, protocols must still be designed carefully.

5. Transports and Semantics

The RPC protocol can be implemented on several different transport protocols. The scope of the definition of the RPC protocol excludes how a message is passed from one process to another, and includes only the specification and interpretation of messages. However, the application may wish to obtain information about (and perhaps control over) the transport layer through an interface not specified in this document. For example, the transport protocol may impose a restriction on the maximum size of RPC messages, or it may be stream-oriented like TCP [RFC0793] with no size limit. The client and server must agree on their transport protocol choices.

It is important to point out that RPC does not try to implement any kind of reliability and that the application may need to be aware of the type of transport protocol underneath RPC. If it knows it is running on top of a reliable transport such as TCP, then most of the work is already done for it. On the other hand, if it is running on
top of an unreliable transport such as UDP [RFC0768], it must implement its own time-out, retransmission, and duplicate detection policies as the RPC protocol does not provide these services.

Because of transport independence, the RPC protocol does not attach specific semantics to the remote procedures or their execution requirements. Semantics can be inferred from (but should be explicitly specified by) the underlying transport protocol. For example, consider RPC running on top of an unreliable transport such as UDP. If an application retransmits RPC call messages after time-outs, and does not receive a reply, it cannot infer anything about the number of times the procedure was executed. If it does receive a reply, then it can infer that the procedure was executed at least once.

A server may wish to remember previously granted requests from a client and not regrant them, in order to insure some degree of execute-at-most-once semantics. A server can do this by taking advantage of the transaction ID that is packaged with every RPC message. The main use of this transaction ID is by the client RPC entity in matching replies to calls. However, a client application may choose to reuse its previous transaction ID when retransmitting a call. The server may choose to remember this ID after executing a call and not execute calls with the same ID, in order to achieve some degree of execute-at-most-once semantics. The server is not allowed to examine this ID in any other way except as a test for equality.

On the other hand, if using a "reliable" transport such as TCP, the application can infer from a reply message that the procedure was executed exactly once, but if it receives no reply message, it cannot assume that the remote procedure was not executed. Note that even if a connection-oriented protocol like TCP is used, an application still needs time-outs and reconnections to handle server crashes.

There are other possibilities for transports besides datagram- or connection-oriented protocols. For example, a request-reply protocol such as [VMTP] is perhaps a natural transport for RPC. ONC RPC currently uses both TCP and UDP transport protocols. Section 11 ("Record Marking Standard") describes the mechanism employed by ONC RPC to utilize a connection-oriented, stream-oriented transport such as TCP. The mechanism by which future transports having different structural characteristics should be used to transfer ONC RPC messages should be specified by means of a Standards Track RFC, once such additional transports are defined.
6. Binding and Rendezvous Independence

The act of binding a particular client to a particular service and transport parameters is NOT part of this RPC protocol specification. This important and necessary function is left up to some higher-level software.

Implementors could think of the RPC protocol as the jump-subroutine instruction (JSR) of a network; the loader (binder) makes JSR useful, and the loader itself uses JSR to accomplish its task. Likewise, the binding software makes RPC useful, possibly using RPC to accomplish this task.

7. Authentication

The RPC protocol provides the fields necessary for a client to identify itself to a service, and vice-versa, in each call and reply message. Security and access control mechanisms can be built on top of this message authentication. Several different authentication protocols can be supported. A field in the RPC header indicates which protocol is being used. More information on specific authentication protocols is in Section 8.2, "Authentication, Integrity and Privacy".

8. RPC Protocol Requirements

The RPC protocol must provide for the following:

- Unique specification of a procedure to be called
- Provisions for matching response messages to request messages
- Provisions for authenticating the caller to service and vice-versa

Besides these requirements, features that detect the following are worth supporting because of protocol roll-over errors, implementation bugs, user error, and network administration:

- RPC protocol mismatches
- Remote program protocol version mismatches
- Protocol errors (such as misspecification of a procedure’s parameters)
- Reasons why remote authentication failed
- Any other reasons why the desired procedure was not called
8.1. RPC Programs and Procedures

The RPC call message has three unsigned-integer fields -- remote program number, remote program version number, and remote procedure number -- that uniquely identify the procedure to be called. Program numbers are administered by a central authority (IANA). Once implementors have a program number, they can implement their remote program; the first implementation would most likely have the version number 1 but MUST NOT be the number zero. Because most new protocols evolve, a "version" field of the call message identifies which version of the protocol the caller is using. Version numbers enable support of both old and new protocols through the same server process.

The procedure number identifies the procedure to be called. These numbers are documented in the specific program's protocol specification. For example, a file service's protocol specification may state that its procedure number 5 is "read" and procedure number 12 is "write".

Just as remote program protocols may change over several versions, the actual RPC message protocol could also change. Therefore, the call message also has in it the RPC version number, which is always equal to 2 for the version of RPC described here.

The reply message to a request message has enough information to distinguish the following error conditions:

- The remote implementation of RPC does not support protocol version 2. The lowest and highest supported RPC version numbers are returned.
- The remote program is not available on the remote system.
- The remote program does not support the requested version number. The lowest and highest supported remote program version numbers are returned.
- The requested procedure number does not exist. (This is usually a client-side protocol or programming error.)
- The parameters to the remote procedure appear to be garbage from the server's point of view. (Again, this is usually caused by a disagreement about the protocol between client and service.)
8.2. Authentication, Integrity, and Privacy

Provisions for authentication of caller to service and vice-versa are provided as a part of the RPC protocol. The call message has two authentication fields: the credential and the verifier. The reply message has one authentication field: the response verifier. The RPC protocol specification defines all three fields to be the following opaque type (in the eXternal Data Representation (XDR) language [RFC4506]):

```c
enum auth_flavor {
    AUTH_NONE       = 0,
    AUTH_SYS        = 1,
    AUTH_SHORT      = 2,
    AUTH_DH         = 3,
    RPCSEC_GSS      = 6
    /* and more to be defined */
};

struct opaque_auth {
    auth_flavor flavor;
    opaque body<400>;
};
```

In other words, any "opaque_auth" structure is an "auth_flavor" enumeration followed by up to 400 bytes that are opaque to (uninterpreted by) the RPC protocol implementation.

The interpretation and semantics of the data contained within the authentication fields are specified by individual, independent authentication protocol specifications.

If authentication parameters were rejected, the reply message contains information stating why they were rejected.

As demonstrated by RPCSEC_GSS, it is possible for an "auth_flavor" to also support integrity and privacy.
8.3. Program Number Assignment

Program numbers are given out in groups according to the following chart:

- **Reserved**
- **To be assigned by IANA**
- **Defined by local administrator** (some blocks assigned here)
- **Transient**
- **Reserved**
- **Assignment outstanding**
- **Reserved**

The first group is a range of numbers administered by IANA and should be identical for all sites. The second range is for applications peculiar to a particular site. This range is intended primarily for debugging new programs. When a site develops an application that might be of general interest, that application should be given an assigned number in the first range. Application developers may apply for blocks of RPC program numbers in the first range by methods described in Appendix B. The third group is for applications that generate program numbers dynamically. The final groups are reserved for future use, and should not be used.

8.4. Other Uses of the RPC Protocol

The intended use of this protocol is for calling remote procedures. Normally, each call message is matched with a reply message. However, the protocol itself is a message-passing protocol with which other (non-procedure-call) protocols can be implemented.

8.4.1. Batching

Batching is useful when a client wishes to send an arbitrarily large sequence of call messages to a server. Batching typically uses reliable byte stream protocols (like TCP) for its transport. In the case of batching, the client never waits for a reply from the server, and the server does not send replies to batch calls. A sequence of batch calls is usually terminated by a legitimate remote procedure call operation in order to flush the pipeline and get positive acknowledgement.
8.4.2. Broadcast Remote Procedure Calls

In broadcast protocols, the client sends a broadcast call to the network and waits for numerous replies. This requires the use of packet-based protocols (like UDP) as its transport protocol. Servers that support broadcast protocols usually respond only when the call is successfully processed and are silent in the face of errors, but this varies with the application.

The principles of broadcast RPC also apply to multicasting -- an RPC request can be sent to a multicast address.

9. The RPC Message Protocol

This section defines the RPC message protocol in the XDR data description language [RFC4506].

```c
enum msg_type {
    CALL = 0,
    REPLY = 1
};
```

A reply to a call message can take on two forms: the message was either accepted or rejected.

```c
enum reply_stat {
    MSG_ACCEPTED = 0,
    MSG_DENIED   = 1
};
```

Given that a call message was accepted, the following is the status of an attempt to call a remote procedure.

```c
enum accept_stat {
    SUCCESS       = 0, /* RPC executed successfully */
    PROG_UNAVAIL  = 1, /* remote hasn’t exported program */
    PROG_MISMATCH = 2, /* remote can’t support version # */
    PROC_UNAVAIL  = 3, /* program can’t support procedure */
    GARBAGE_ARGS  = 4, /* procedure can’t decode params */
    SYSTEM_ERR    = 5  /* e.g. memory allocation failure */
};
```

Reasons why a call message was rejected:

```c
enum reject_stat {
    RPC_MISMATCH = 0, /* RPC version number != 2 */
    AUTH_ERROR = 1    /* remote can’t authenticate caller */
};
```
Why authentication failed:

```c
enum auth_stat {
    AUTH_OK             = 0,  /* success                        */
    /*
     * failed at remote end
     */
    AUTH_BADCRED        = 1,  /* bad credential (seal broken)   */
    AUTH_REJECTEDCRED   = 2,  /* client must begin new session  */
    AUTH_BADVERF        = 3,  /* bad verifier (seal broken)     */
    AUTH_REJECTEDVERF   = 4,  /* verifier expired or replayed   */
    AUTH_TOOWEAK         = 5,  /* rejected for security reasons */
    /*
     * failed locally
     */
    AUTH_INVALIDRESP    = 6,  /* bogus response verifier        */
    AUTH_FAILED         = 7,  /* reason unknown                 */
    /*
     * AUTH_KERB errors; deprecated.  See [RFC2695]
     */
    AUTH_KERB_GENERIC   = 8,  /* kerberos generic error         */
    AUTH_TIMEEXPIRE     = 9,  /* time of credential expired     */
    AUTH_TKT_FILE       = 10, /* problem with ticket file       */
    AUTH_DECODE         = 11, /* can’t decode authenticator      */
    AUTH_NET_ADDR       = 12, /* wrong net address in ticket     */
    /*
     * RPCSEC_GSS GSS related errors
     */
    RPCSEC_GSS_CREDPROBLEM = 13, /* no credentials for user        */
    RPCSEC_GSS_CTXPROBLEM = 14   /* problem with context           */
};
```

As new authentication mechanisms are added, there may be a need for more status codes to support them. IANA will hand out new auth_stat numbers on a simple First Come First Served basis as defined in the "IANA Considerations" and Appendix B.

The RPC message:

All messages start with a transaction identifier, xid, followed by a two-armed discriminated union. The union’s discriminant is a msg_type that switches to one of the two types of the message. The xid of a REPLY message always matches that of the initiating CALL message. NB: The "xid" field is only used for clients matching reply messages with call messages or for servers detecting retransmissions; the service side cannot treat this id as any type of sequence number.
struct rpc_msg {
    unsigned int xid;
    union switch (msg_type mtype) {
        case CALL:
            call_body cbody;
        case REPLY:
            reply_body rbody;
    } body;
};

Body of an RPC call:

In version 2 of the RPC protocol specification, rpcvers MUST be equal
to 2. The fields "prog", "vers", and "proc" specify the remote
program, its version number, and the procedure within the remote
program to be called. After these fields are two authentication
parameters: cred (authentication credential) and verf (authentication
verifier). The two authentication parameters are followed by the
parameters to the remote procedure, which are specified by the
specific program protocol.

The purpose of the authentication verifier is to validate the
authentication credential. Note that these two items are
historically separate, but are always used together as one logical
entity.

struct call_body {
    unsigned int rpcvers;       /* must be equal to two (2) */
    unsigned int prog;
    unsigned int vers;
    unsigned int proc;
    opaque_auth cred;
    opaque_auth verf;
    /* procedure-specific parameters start here */
};

Body of a reply to an RPC call:

union reply_body switch (reply_stat stat) {
    case MSG_ACCEPTED:
        accepted_reply areply;
    case MSG_DENIED:
        rejected_reply rreply;
} reply;
Reply to an RPC call that was accepted by the server:

There could be an error even though the call was accepted. The first field is an authentication verifier that the server generates in order to validate itself to the client. It is followed by a union whose discriminant is an enum accept_stat. The SUCCESS arm of the union is protocol-specific. The PROG_UNAVAIL, PROC_UNAVAIL, GARBAGE_ARGS, and SYSTEM_ERR arms of the union are void. The PROG_MISMATCH arm specifies the lowest and highest version numbers of the remote program supported by the server.

```c
struct accepted_reply {
    opaque_auth verf;
    union switch (accept_stat stat) {
        case SUCCESS:
            opaque results[0];
            /*
             * procedure-specific results start here
             */
        case PROG_MISMATCH:
            struct {
                unsigned int low;
                unsigned int high;
            } mismatch_info;
        default:
            /*
             * Void. Cases include PROG_UNAVAIL, PROC_UNAVAIL,
             * GARBAGE_ARGS, and SYSTEM_ERR.
             */
            void;
    } reply_data;
};
```

Reply to an RPC call that was rejected by the server:

The call can be rejected for two reasons: either the server is not running a compatible version of the RPC protocol (RPC_MISMATCH) or the server rejects the identity of the caller (AUTH_ERROR). In case of an RPC version mismatch, the server returns the lowest and highest supported RPC version numbers. In case of invalid authentication, failure status is returned.
union rejected_reply switch (reject_stat stat) {
  case RPC_MISMATCH:
    struct {
      unsigned int low;
      unsigned int high;
    } mismatch_info;
  case AUTH_ERROR:
    auth_stat stat;
};

10. Authentication Protocols

As previously stated, authentication parameters are opaque, but open-ended to the rest of the RPC protocol. This section defines two standard flavors of authentication. Implementors are free to invent new authentication types, with the same rules of flavor number assignment as there are for program number assignment. The flavor of a credential or verifier refers to the value of the "flavor" field in the opaque_auth structure. Flavor numbers, like RPC program numbers, are also administered centrally, and developers may assign new flavor numbers by methods described in Appendix B. Credentials and verifiers are represented as variable-length opaque data (the "body" field in the opaque_auth structure).

In this document, two flavors of authentication are described. Of these, Null authentication (described in the next subsection) is mandatory -- it MUST be available in all implementations. System authentication (AUTH_SYS) is described in Appendix A. Implementors MAY include AUTH_SYS in their implementations to support existing applications. See "Security Considerations" for information about other, more secure, authentication flavors.

10.1. Null Authentication

Often, calls must be made where the client does not care about its identity or the server does not care who the client is. In this case, the flavor of the RPC message's credential, verifier, and reply verifier is "AUTH_NONE". Opaque data associated with "AUTH_NONE" is undefined. It is recommended that the length of the opaque data be zero.
11. Record Marking Standard

When RPC messages are passed on top of a byte stream transport protocol (like TCP), it is necessary to delimit one message from another in order to detect and possibly recover from protocol errors. This is called record marking (RM). One RPC message fits into one RM record.

A record is composed of one or more record fragments. A record fragment is a four-byte header followed by 0 to \((2^{31}) - 1\) bytes of fragment data. The bytes encode an unsigned binary number; as with XDR integers, the byte order is from highest to lowest. The number encodes two values -- a boolean that indicates whether the fragment is the last fragment of the record (bit value 1 implies the fragment is the last fragment) and a 31-bit unsigned binary value that is the length in bytes of the fragment’s data. The boolean value is the highest-order bit of the header; the length is the 31 low-order bits. (Note that this record specification is NOT in XDR standard form!)

12. The RPC Language

Just as there was a need to describe the XDR data-types in a formal language, there is also need to describe the procedures that operate on these XDR data-types in a formal language as well. The RPC language is an extension to the XDR language, with the addition of "program", "procedure", and "version" declarations. The keywords "program" and "version" are reserved in the RPC language, and implementations of XDR compilers MAY reserve these keywords even when provided with pure XDR, non-RPC, descriptions. The following example is used to describe the essence of the language.
12.1. An Example Service Described in the RPC Language

Here is an example of the specification of a simple ping program.

```c
program PING_PROG {
  /*
   * Latest and greatest version
   */
  version PING_VERS_PINGBACK {
    void
    PINGPROC_NULL(void) = 0;
    /*
     * Ping the client, return the round-trip time
     * (in microseconds). Returns -1 if the operation
     * timed out.
     */
    int
    PINGPROC_PINGBACK(void) = 1;
  } = 2;

  /*
   * Original version
   */
  version PING_VERS_ORIG {
    void
    PINGPROC_NULL(void) = 0;
  } = 1;
}

const PING_VERS = 2;  /* latest version */
```

The first version described is PING_VERS_PINGBACK with two procedures: PINGPROC_NULL and PINGPROC_PINGBACK. PINGPROC_NULL takes no arguments and returns no results, but it is useful for computing round-trip times from the client to the server and back again. By convention, procedure 0 of any RPC protocol should have the same semantics and never require any kind of authentication. The second procedure is used for the client to have the server do a reverse ping operation back to the client, and it returns the amount of time (in microseconds) that the operation used. The next version, PING_VERS_ORIG, is the original version of the protocol, and it does not contain the PINGPROC_PINGBACK procedure. It is useful for compatibility with old client programs, and as this program matures, it may be dropped from the protocol entirely.
12.2. The RPC Language Specification

The RPC language is identical to the XDR language defined in RFC 4506, except for the added definition of a "program-def", described below.

program-def:
   "program" identifier "{" version-def version-def *
   "}" "=" constant ";"

version-def:
   "version" identifier "{" procedure-def procedure-def *
   "}" "=" constant ";"

procedure-def:
   proc-return identifier "(" proc-firstarg ("," type-specifier )* "")" "=" constant ";"

proc-return: "void" | type-specifier
proc-firstarg: "void" | type-specifier

12.3. Syntax Notes

- The following keywords are added and cannot be used as identifiers: "program" and "version".

- A version name cannot occur more than once within the scope of a program definition. Neither can a version number occur more than once within the scope of a program definition.

- A procedure name cannot occur more than once within the scope of a version definition. Neither can a procedure number occur more than once within the scope of version definition.

- Program identifiers are in the same name space as constant and type identifiers.

- Only unsigned constants can be assigned to programs, versions, and procedures.

- Current RPC language compilers do not generally support more than one type-specifier in procedure argument lists; the usual practice is to wrap arguments into a structure.
13. IANA Considerations

The assignment of RPC program numbers, authentication flavor numbers, and authentication status numbers has in the past been performed by Sun Microsystems, Inc (Sun). This is inappropriate for an IETF Standards Track protocol, as such work is done well by the Internet Assigned Numbers Authority (IANA). This document proposes the transfer of authority over RPC program numbers, authentication flavor numbers, and authentication status numbers described here from Sun Microsystems, Inc. to IANA and describes how IANA will maintain and assign these numbers. Users of RPC protocols will benefit by having an independent body responsible for these number assignments.

13.1. Numbering Requests to IANA

Appendix B of this document describes the information to be sent to IANA to request one or more RPC numbers and the rules that apply. IANA will store the request for documentary purposes and put the following information into the public registry:

- The short description of purpose and use
- The program number(s) assigned
- The short identifier string(s)

13.2. Protecting Past Assignments

Sun has made assignments in both the RPC program number space and the RPC authentication flavor number space since the original deployment of RPC. The assignments made by Sun Microsystems are still valid, and will be preserved. Sun has communicated all current assignments in both number spaces to IANA and final handoff of number assignment is complete. Current program and auth number assignments are provided in Appendix C. Current authentication status numbers are listed in Section 9 of this document in the "enum auth_stat" definition.

13.3. RPC Number Assignment

Future IANA practice will deal with the following partitioning of the 32-bit number space as listed in Section 8.3. Detailed information for the administration of the partitioned blocks in Section 8.3 is given below.
13.3.1. To Be Assigned By IANA

The first block will be administered by IANA, with previous assignments by Sun protected. Previous assignments were restricted to the range decimal 100000-399999 (0x000186a0 to 0x00061a7f); therefore, IANA will begin assignments at decimal 400000. Individual numbers should be granted on a First Come First Served basis, and blocks should be granted under rules related to the size of the block.

13.3.2. Defined by Local Administrator

The "Defined by local administrator" block is available for any local administrative domain to use, in a similar manner to IP address ranges reserved for private use. The expected use would be through the establishment of a local domain "authority" for assigning numbers from this range. This authority would establish any policies or procedures to be used within that local domain for use or assignment of RPC numbers from the range. The local domain should be sufficiently isolated that it would be unlikely that RPC applications developed by other local domains could communicate with the domain. This could result in RPC number contention, which would cause one of the applications to fail. In the absence of a local administrator, this block can be utilized in a "Private Use" manner per [RFC5226].

13.3.3. Transient Block

The "Transient" block can be used by any RPC application on an "as available" basis. This range is intended for services that can communicate a dynamically selected RPC program number to clients of the service. Any mechanism can be used to communicate the number. For example, either shared memory when the client and server are located on the same system or a network message (either RPC or otherwise) that disseminates the selected number can be used.

The transient block is not administered. An RPC service uses this range by selecting a number in the transient range and attempting to register that number with the local system’s RPC bindery (see the RPCBPROC_SET or PMAPPROC_SET procedures in "Binding Protocols for ONC RPC Version 2", [RFC1833]). If successful, no other RPC service was using that number and the RPC Bindery has assigned that number to the requesting RPC application. The registration is valid until the RPC Bindery terminates, which normally would only happen if the system reboots, causing all applications, including the RPC service using the transient number, to terminate. If the transient number registration fails, another RPC application is using the number and...
the requestor must select another number and try again. To avoid conflicts, the recommended method is to select a number randomly from the transient range.

13.3.4. Reserved Block

The "Reserved" blocks are available for future use. RPC applications must not use numbers in these ranges unless their use is allowed by future action by the IESG.

13.3.5. RPC Number Sub-Blocks

RPC numbers are usually assigned for specific RPC services. Some applications, however, require multiple RPC numbers for a service. The most common example is an RPC service that needs to have multiple instances of the service active simultaneously at a specific site. RPC does not have an "instance identifier" in the protocol, so either a mechanism must be implemented to multiplex RPC requests amongst various instances of the service or unique RPC numbers must be used by each instance.

In these cases, the RPC protocol used with the various numbers may be different or the same. The numbers may either be assigned dynamically by the application, or as part of a site-specific administrative decision. If possible, RPC services that dynamically assign RPC numbers should use the "Transient" RPC number block defined in Section 13.3.3. If not possible, RPC number sub-blocks may be requested.

Assignment of RPC Number Sub-Blocks is controlled by the size of the sub-block being requested. "Specification Required" and "IESG Approval" are used as defined by Section 4.1 of [RFC5226].

<table>
<thead>
<tr>
<th>Size of sub-block</th>
<th>Assignment Method</th>
<th>Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to 100 numbers</td>
<td>First Come First Served</td>
<td>IANA</td>
</tr>
<tr>
<td>Up to 1000 numbers</td>
<td>Specification Required</td>
<td>IANA</td>
</tr>
<tr>
<td>More than 1000 numbers</td>
<td>IESG Approval required</td>
<td>IESG</td>
</tr>
</tbody>
</table>

Note: sub-blocks can be any size. The limits given above are maximums, and smaller size sub-blocks are allowed.

Sub-blocks sized up to 100 numbers may be assigned by IANA on a First Come First Served basis. The RPC Service Description included in the range must include an indication of how the sub-block is managed. At a minimum, the statement should indicate whether the sub-block is
used with a single RPC protocol or multiple RPC protocols, and whether the numbers are dynamically assigned or statically (through administrative action) assigned.

Sub-blocks of up to 1000 numbers must be documented in detail. The documentation must describe the RPC protocol or protocols that are to be used in the range. It must also describe how the numbers within the sub-block are to be assigned or used.

Sub-blocks sized over 1000 numbers must be documented as described above, and the assignment must be approved by the IESG. It is expected that this will be rare.

In order to avoid multiple requests of large blocks of numbers, the following rule is proposed.

Requests up to and including 100 RPC numbers are handled via the First Come First Served assignment method. This 100 number threshold applies to the total number of RPC numbers assigned to an individual or entity. For example, if an individual or entity first requests, say, 70 numbers, and then later requests 40 numbers, then the request for the 40 numbers will be assigned via the Specification Required method. As long as the total number of numbers assigned does not exceed 1000, IANA is free to waive the Specification Required assignment for incremental requests of less than 100 numbers.

If an individual or entity has under 1000 numbers and later requests an additional set of numbers such that the individual or entity would be granted over 1000 numbers, then the additional request will require IESG Approval.

13.4. RPC Authentication Flavor Number Assignment

The second number space is the authentication mechanism identifier, or "flavor", number. This number is used to distinguish between various authentication mechanisms that can be optionally used with an RPC message. An authentication identifier is used in the "flavor" field of the "opaque_auth" structure.

13.4.1. Assignment Policy

Appendix B of this document describes the information to be sent to IANA to request one or more RPC auth numbers and the rules that apply. IANA will store the request for documentary purposes and put the following information into the public registry:
13.4.2. Auth Flavors vs. Pseudo-Flavors

Recent progress in RPC security has moved away from new auth flavors as used by AUTH_DH [DH], and has focused on using the existing RPCSEC_GSS [RFC2203] flavor and inventing novel GSS-API (Generic Security Services Application Programming Interface) mechanisms that can be used with it. Even though RPCSEC_GSS is an assigned authentication flavor, use of a new RPCSEC_GSS mechanism with the Network File System (NFS) ([RFC1094] [RFC1813], and [RFC3530]) will require the registration of ‘pseudo-flavors’ that are used to negotiate security mechanisms in an unambiguous way, as defined by [RFC2623]. Existing pseudo-flavors have been granted in the decimal range 390000-390255. New pseudo-flavor requests will be granted by IANA within this block on a First Come First Served basis.

For non-pseudo-flavor requests, IANA will begin granting RPC authentication flavor numbers at 400000 on a First Come First Served basis to avoid conflicts with currently granted numbers.

For authentication flavors or RPCSEC_GSS mechanisms to be used on the Internet, it is strongly advised that an Informational or Standards Track RFC be published describing the authentication mechanism behaviour and parameters.

13.5. Authentication Status Number Assignment

The final number space is the authentication status or "auth_stat" values that describe the nature of a problem found during an attempt to authenticate or validate authentication. The complete initial list of these values is found in Section 9 of this document, in the "auth_stat" enum listing. It is expected that it will be rare to add values, but that a small number of new values may be added from time to time as new authentication flavors introduce new possibilities. Numbers should be granted on a First Come First Served basis to avoid conflicts with currently granted numbers.

13.5.1. Assignment Policy

Appendix B of this document describes the information to be sent to IANA to request one or more auth_stat values and the rules that apply. IANA will store the request for documentary purposes, and put the following information into the public registry:
14. Security Considerations

AUTH_SYS as described in Appendix A is known to be insecure due to the lack of a verifier to permit the credential to be validated. AUTH_SYS SHOULD NOT be used for services that permit clients to modify data. AUTH_SYS MUST NOT be specified as RECOMMENDED or REQUIRED for any Standards Track RPC service.

AUTH_DH as mentioned in Sections 8.2 and 13.4.2 is considered obsolete and insecure; see [RFC2695]. AUTH_DH SHOULD NOT be used for services that permit clients to modify data. AUTH_DH MUST NOT be specified as RECOMMENDED or REQUIRED for any Standards Track RPC service.

[RFC2203] defines a new security flavor, RPCSEC_GSS, which permits GSS-API [RFC2743] mechanisms to be used for securing RPC. All non-trivial RPC programs developed in the future should implement RPCSEC_GSS-based security appropriately. [RFC2623] describes how this was done for a widely deployed RPC program.

Standards Track RPC services MUST mandate support for RPCSEC_GSS, and MUST mandate support for an authentication pseudo-flavor with appropriate levels of security, depending on the need for simple authentication, integrity (a.k.a. non-repudiation), or data privacy.
Appendix A: System Authentication

The client may wish to identify itself, for example, as it is identified on a UNIX system. The flavor of the client credential is "AUTH_SYS". The opaque data constituting the credential encodes the following structure:

```c
struct authsys_parms {
    unsigned int stamp;
    string machinename<255>;
    unsigned int uid;
    unsigned int gid;
    unsigned int gids<16>;
};
```

The "stamp" is an arbitrary ID that the caller machine may generate. The "machinename" is the name of the caller's machine (like "krypton"). The "uid" is the caller's effective user ID. The "gid" is the caller's effective group ID. "gids" are a counted array of groups that contain the caller as a member. The verifier accompanying the credential should have "AUTH_NONE" flavor value (defined above). Note that this credential is only unique within a particular domain of machine names, uids, and gids.

The flavor value of the verifier received in the reply message from the server may be "AUTH_NONE" or "AUTH_SHORT". In the case of "AUTH_SHORT", the bytes of the reply verifier's string encode an opaque structure. This new opaque structure may now be passed to the server instead of the original "AUTH_SYS" flavor credential. The server may keep a cache that maps shorthand opaque structures (passed back by way of an "AUTH_SHORT" style reply verifier) to the original credentials of the caller. The caller can save network bandwidth and server cpu cycles by using the shorthand credential.

The server may flush the shorthand opaque structure at any time. If this happens, the remote procedure call message will be rejected due to an authentication error. The reason for the failure will be "AUTH_REJECTEDCRED". At this point, the client may wish to try the original "AUTH_SYS" style of credential.

It should be noted that use of this flavor of authentication does not guarantee any security for the users or providers of a service, in itself. The authentication provided by this scheme can be considered legitimate only when applications using this scheme and the network can be secured externally, and privileged transport addresses are used for the communicating end-points (an example of this is the use of privileged TCP/UDP ports in UNIX systems -- note that not all systems enforce privileged transport address mechanisms).
Appendix B: Requesting RPC-Related Numbers from IANA

RPC program numbers, authentication flavor numbers, and authentication status numbers that must be unique across all networks are assigned by the Internet Assigned Number Authority. To apply for a single number or a block of numbers, electronic mail must be sent to IANA <iana@iana.org> with the following information:

- The type of number(s) (program number or authentication flavor number or authentication status number) sought
- How many numbers are sought
- The name of the person or company that will use the number
- An "identifier string" that associates the number with a service
- Email address of the contact person for the service that will be using the number
- A short description of the purpose and use of the number
- If an authentication flavor number is sought, and the number will be a ‘pseudo-flavor’ intended for use with RPCSEC_GSS and NFS, mappings analogous to those in Section 4.2 of [RFC2623]

Specific numbers cannot be requested. Numbers are assigned on a First Come First Served basis.

For all RPC authentication flavor and authentication status numbers to be used on the Internet, it is strongly advised that an Informational or Standards Track RFC be published describing the authentication mechanism behaviour and parameters.
## Appendix C: Current Number Assignments

<table>
<thead>
<tr>
<th>Description/Owner</th>
<th>RPC Program Number</th>
<th>Short Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>portmapper</td>
<td>100000</td>
<td>pmapprog portmap rpcbind</td>
</tr>
<tr>
<td>remote stats</td>
<td>100001</td>
<td>rstatprog</td>
</tr>
<tr>
<td>remote users</td>
<td>100002</td>
<td>rusersprog</td>
</tr>
<tr>
<td>nfs</td>
<td>100003</td>
<td>nfs</td>
</tr>
<tr>
<td>yellow pages (NIS)</td>
<td>100004</td>
<td>ypprog ypserv</td>
</tr>
<tr>
<td>mount demon</td>
<td>100005</td>
<td>mountprog</td>
</tr>
<tr>
<td>remote dbx</td>
<td>100006</td>
<td>dbxprog</td>
</tr>
<tr>
<td>yp binder (NIS)</td>
<td>100007</td>
<td>ypbindingpro ypbind</td>
</tr>
<tr>
<td>shutdown msg</td>
<td>100008</td>
<td>wall</td>
</tr>
<tr>
<td>yppasswd server</td>
<td>100009</td>
<td>yppasswdpro yppasswdd</td>
</tr>
<tr>
<td>ether stats</td>
<td>100010</td>
<td>etherstatpro</td>
</tr>
<tr>
<td>disk quotas</td>
<td>100011</td>
<td>rquota</td>
</tr>
<tr>
<td>spray packets</td>
<td>100012</td>
<td>spray</td>
</tr>
<tr>
<td>3270 mapper</td>
<td>100013</td>
<td>ibm3270pro</td>
</tr>
<tr>
<td>RJE mapper</td>
<td>100014</td>
<td>ibmrjepro</td>
</tr>
<tr>
<td>selection service</td>
<td>100015</td>
<td>selnsvcpro</td>
</tr>
<tr>
<td>remote database access</td>
<td>100016</td>
<td>rdatabasepro</td>
</tr>
<tr>
<td>remote execution</td>
<td>100017</td>
<td>rexec</td>
</tr>
<tr>
<td>Alice Office Automation</td>
<td>100018</td>
<td>alicepro</td>
</tr>
<tr>
<td>scheduling service</td>
<td>100019</td>
<td>schedprog</td>
</tr>
<tr>
<td>local lock manager</td>
<td>100020</td>
<td>lockprog llockmgr</td>
</tr>
<tr>
<td>network lock manager</td>
<td>100021</td>
<td>netlockprog nlockmgr</td>
</tr>
<tr>
<td>x.25 inr protocol</td>
<td>100022</td>
<td>x25prog</td>
</tr>
<tr>
<td>status monitor 1</td>
<td>100023</td>
<td>statmon1</td>
</tr>
<tr>
<td>status monitor 2</td>
<td>100024</td>
<td>statmon2</td>
</tr>
<tr>
<td>selection library</td>
<td>100025</td>
<td>selnlibpro</td>
</tr>
<tr>
<td>boot parameters service</td>
<td>100026</td>
<td>bootparam</td>
</tr>
<tr>
<td>mazewars game</td>
<td>100027</td>
<td>mazeaprog</td>
</tr>
<tr>
<td>yp update (NIS)</td>
<td>100028</td>
<td>ypupdatepro ypupdate</td>
</tr>
<tr>
<td>key server</td>
<td>100029</td>
<td>keyservepro</td>
</tr>
<tr>
<td>secure login</td>
<td>100030</td>
<td>securecmdprog</td>
</tr>
<tr>
<td>nfs net forwarder init</td>
<td>100031</td>
<td>netfwdiprog</td>
</tr>
<tr>
<td>nfs net forwarder trans</td>
<td>100032</td>
<td>netfwdtprog</td>
</tr>
<tr>
<td>sunlink MAP</td>
<td>100033</td>
<td>sunlinkmap</td>
</tr>
<tr>
<td>network monitor</td>
<td>100034</td>
<td>netmonpro</td>
</tr>
<tr>
<td>lightweight database</td>
<td>100035</td>
<td>dbasepro</td>
</tr>
<tr>
<td>password authorization</td>
<td>100036</td>
<td>pwdauthpro</td>
</tr>
<tr>
<td>translucent file svc</td>
<td>100037</td>
<td>tfsprog</td>
</tr>
<tr>
<td>nse server</td>
<td>100038</td>
<td>nseprog</td>
</tr>
<tr>
<td>nse activate daemon</td>
<td>100039</td>
<td>nse_activate_prog</td>
</tr>
<tr>
<td>sunview help</td>
<td>100040</td>
<td>sunview_help_progs</td>
</tr>
<tr>
<td>Service Description</td>
<td>100041</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>pnp install</td>
<td>pnp_prog</td>
<td></td>
</tr>
<tr>
<td>ip addr allocator</td>
<td>ipaddr_alloc_prog</td>
<td></td>
</tr>
<tr>
<td>show filehandle</td>
<td>filehandle</td>
<td></td>
</tr>
<tr>
<td>MVS NFS mount</td>
<td>mvsnsfsprog</td>
<td></td>
</tr>
<tr>
<td>remote user file operations</td>
<td>rem_fileop_user_prog</td>
<td></td>
</tr>
<tr>
<td>batched ypupdate</td>
<td>batch_ypupdateprog</td>
<td></td>
</tr>
<tr>
<td>network execution mgr</td>
<td>nem_prog</td>
<td></td>
</tr>
<tr>
<td>raytrace/mandelbrot remote daemon</td>
<td>raytrace_rd_prog</td>
<td></td>
</tr>
<tr>
<td>raytrace/mandelbrot local daemon</td>
<td>raytrace_ld_prog</td>
<td></td>
</tr>
<tr>
<td>remote group file operations</td>
<td>rem_fileop_group_prog</td>
<td></td>
</tr>
<tr>
<td>remote system file operations</td>
<td>rem_fileop_system_prog</td>
<td></td>
</tr>
<tr>
<td>remote system role operations</td>
<td>rem_system_role_prog</td>
<td></td>
</tr>
<tr>
<td>gpd lego fb simulator</td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>gpd simulator interface</td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>ioadmd</td>
<td>ioadmd</td>
<td></td>
</tr>
<tr>
<td>filemerge</td>
<td>filemerge_prog</td>
<td></td>
</tr>
<tr>
<td>Name Binding Program</td>
<td>namebind_prog</td>
<td></td>
</tr>
<tr>
<td>sunlink NJE</td>
<td>njeprog</td>
<td></td>
</tr>
<tr>
<td>MVS NSF/S get attribute service</td>
<td>mvsattrprog</td>
<td></td>
</tr>
<tr>
<td>SunAccess/SunLink resource manager</td>
<td>rmgrprog</td>
<td></td>
</tr>
<tr>
<td>UID allocation service</td>
<td>uidallocprog</td>
<td></td>
</tr>
<tr>
<td>license broker</td>
<td>lbserverprog</td>
<td></td>
</tr>
<tr>
<td>NETLicense client binder</td>
<td>lbbinderprog</td>
<td></td>
</tr>
<tr>
<td>GID allocation service</td>
<td>gidallocprog</td>
<td></td>
</tr>
<tr>
<td>SunIsam</td>
<td>sunisamprog</td>
<td></td>
</tr>
<tr>
<td>Remote Debug Server</td>
<td>rdbsrvprog</td>
<td></td>
</tr>
<tr>
<td>Network Directory Daemon</td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>Network Calendar Program</td>
<td>cm is cm</td>
<td></td>
</tr>
<tr>
<td>ypxfrd</td>
<td>ypxfrd</td>
<td></td>
</tr>
<tr>
<td>rpc.timed</td>
<td>timedprog</td>
<td></td>
</tr>
<tr>
<td>bugtraqd</td>
<td>bugtraqقدم</td>
<td></td>
</tr>
<tr>
<td>Connectathon Billboard - NFS</td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>Connectathon Billboard - X</td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>Sun tool for scheduling rooms</td>
<td>schedroom</td>
<td></td>
</tr>
<tr>
<td>Authentication Negotiation</td>
<td>authnegotiate_prog</td>
<td></td>
</tr>
<tr>
<td>Database manipulation</td>
<td>attribute_prog</td>
<td></td>
</tr>
<tr>
<td>Kerberos authentication daemon</td>
<td>kerbprog</td>
<td></td>
</tr>
<tr>
<td>Internal testing product (no name)</td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>Sun Consulting Special</td>
<td>autodump_prog</td>
<td></td>
</tr>
<tr>
<td>Event protocol</td>
<td>event_svc</td>
<td></td>
</tr>
<tr>
<td>bugtraqقدم</td>
<td>bugtraqقدم</td>
<td></td>
</tr>
<tr>
<td>ToolTalk and Link Service Project</td>
<td>database service</td>
<td></td>
</tr>
<tr>
<td>Consulting Services</td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>Consulting Services</td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>Consulting Services</td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>Jupiter Administration</td>
<td>adm_agent admind</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[unknown]</td>
<td></td>
</tr>
<tr>
<td>Service Description</td>
<td>Port Number</td>
<td>Comment</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Dual Disk support</td>
<td>100089</td>
<td>[unknown]</td>
</tr>
<tr>
<td>DocViewer 1.1</td>
<td>100090</td>
<td>libdsd/dsd</td>
</tr>
<tr>
<td>ToolTalk</td>
<td>100091</td>
<td>[unknown]</td>
</tr>
<tr>
<td>Consulting Services</td>
<td>100093</td>
<td>host_checking</td>
</tr>
<tr>
<td>SNA peer-to-peer</td>
<td>100094</td>
<td>[unknown]</td>
</tr>
<tr>
<td>Roger Riggs</td>
<td>100095</td>
<td>searchit</td>
</tr>
<tr>
<td>Robert Allen</td>
<td>100096</td>
<td>mesgtool</td>
</tr>
<tr>
<td>SNA</td>
<td>100097</td>
<td>[unknown]</td>
</tr>
<tr>
<td>SISU</td>
<td>100098</td>
<td>networked version of CS5</td>
</tr>
<tr>
<td>NFS Automat File System</td>
<td>100099</td>
<td>autosf</td>
</tr>
<tr>
<td></td>
<td>100100</td>
<td>msgboard</td>
</tr>
<tr>
<td>event dispatching agent [eventd]</td>
<td>100101</td>
<td>netmgt_eventd_prog</td>
</tr>
<tr>
<td>statistics/event logger [netlogd]</td>
<td>100102</td>
<td>netmgt_netlogd_prog</td>
</tr>
<tr>
<td>topology display manager [topology]</td>
<td>100103</td>
<td>netmgt_topology_prog</td>
</tr>
<tr>
<td>syncstat agent [syncstatd]</td>
<td>100104</td>
<td>netmgt_syncstatd_prog</td>
</tr>
<tr>
<td>ip packet stats agent [ippktd]</td>
<td>100105</td>
<td>netmgt_ippktd_prog</td>
</tr>
<tr>
<td>netmgt config agent [configd]</td>
<td>100106</td>
<td>netmgt_configd_prog</td>
</tr>
<tr>
<td>restat agent [restatd]</td>
<td>100107</td>
<td>netmgt_restatd_prog</td>
</tr>
<tr>
<td>lprg agent [lprstatd]</td>
<td>100108</td>
<td>netmgt_lprstatd_prog</td>
</tr>
<tr>
<td>netmgt activity agent [mgtlogd]</td>
<td>100109</td>
<td>netmgt_mgtlogd_prog</td>
</tr>
<tr>
<td>proxy DECnet NCP agent [proxydni]</td>
<td>100110</td>
<td>netmgt_proxydni_prog</td>
</tr>
<tr>
<td>topology mapper agent [mapperd]</td>
<td>100111</td>
<td>netmgt_mapperd_prog</td>
</tr>
<tr>
<td>netstat agent [netstatd]</td>
<td>100112</td>
<td>netmgt_netstatd_prog</td>
</tr>
<tr>
<td>sample netmgt agent [sampled]</td>
<td>100113</td>
<td>netmgt_sampled_prog</td>
</tr>
<tr>
<td>X.25 statistics agent [vcstatd]</td>
<td>100114</td>
<td>netmgt_vcstatd_prog</td>
</tr>
<tr>
<td>Frame Relay</td>
<td>100128</td>
<td>[unknown]</td>
</tr>
<tr>
<td>PPF agent</td>
<td>100129</td>
<td>[unknown]</td>
</tr>
<tr>
<td>localhad</td>
<td>100130</td>
<td>rpc.localhad</td>
</tr>
<tr>
<td>layers2</td>
<td>100131</td>
<td>na.layers2</td>
</tr>
<tr>
<td>token ring agent</td>
<td>100132</td>
<td>na.tr</td>
</tr>
<tr>
<td>related to lockd and statd</td>
<td>100133</td>
<td>nsm_addr</td>
</tr>
<tr>
<td>Kerberos project</td>
<td>100134</td>
<td>kwarn</td>
</tr>
<tr>
<td>etherif2</td>
<td>100135</td>
<td>na.etherif2</td>
</tr>
<tr>
<td>hostmem2</td>
<td>100136</td>
<td>na.hostmem2</td>
</tr>
<tr>
<td>iostat2</td>
<td>100137</td>
<td>na.iostat2</td>
</tr>
<tr>
<td>snmpv2</td>
<td>100138</td>
<td>na.snmpv2</td>
</tr>
<tr>
<td>Cooperative Console</td>
<td>100139</td>
<td>cc_sender</td>
</tr>
<tr>
<td>na.cpustat</td>
<td>100140</td>
<td>na.cpustat</td>
</tr>
<tr>
<td>Sun Cluster SC3.0</td>
<td>100141</td>
<td>rgmd_receptionist</td>
</tr>
<tr>
<td></td>
<td>100142</td>
<td>fed</td>
</tr>
<tr>
<td>Network Storage</td>
<td>100143</td>
<td>rdc</td>
</tr>
<tr>
<td>Sun Cluster products</td>
<td>100144</td>
<td>nafo</td>
</tr>
<tr>
<td>SunCluster 3.0</td>
<td>100145</td>
<td>scadmd</td>
</tr>
<tr>
<td>ASN.1</td>
<td>100146</td>
<td>amiserv</td>
</tr>
<tr>
<td></td>
<td>100147</td>
<td>amiaux # BER and DER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>encode and decode</td>
</tr>
<tr>
<td>Delegate Management Server</td>
<td>100148</td>
<td>dm</td>
</tr>
</tbody>
</table>
100149  rkstat
100150  ocfser
100151  sccheckd
100152  autoclientd
100153  sunvtsd
100154  ssmond
100155  smser
100156  test1
100157  test2
100158  test3
100159  test4
100160  test5
100161  test6
100162  test7
100163  test8
100164  test9
100165  test10
100166  nfsmapid
100167  SUN_WBEM_C_CIMON_HANDLE
100168  sacmmd
100169  fmd_admin
100170  fmd_api
100171  [unknown]
100172  idmap

unassigned  100173 - 100174
snmptrap  100175  na.snmptrap
unassigned  100176-100199

unassigned  100200
MVS/NFS Memory usage stats server  100201  [unknown]
Netapp  100202-100207
unassigned  100208-100210
8.0 SunLink SNA RJE  100211  [unknown]
8.0 SunLink SNA RJE  100212  [unknown]
  100213  ShowMe
  100214  [unknown]
  100215  [unknown]

AUTH_RSA Key service  100216  keyrsa
SunSelect PC license service  100217  [unknown]
WWCS (Corporate)  100218  sunsolv
  100219  csstatd
X/Open Federated Naming  100220  xfn_server_prog
Kodak Color Management System  100221  kcs_network_io kcs
HA-DBMS  100222  ha_dbms_serv
  100223-100225  [unknown]
  100226  hafaultd

NFS ACL Service  100227  nfs_acl
distributed lock manager  100228  dlmd
100229  metad
100230  metamhd
100231  nfsauth
100232  sadmind
100233  ufsd
100234  grpservd
100235  cachefsd
100236  msmprog Media_Server
100237  ihnamed
100238  ihnetd
100239  ihsecured
100240  ihclassmgrd
100241  ihrepositoryd
100242  metamedd rpc.metamedd
100243  contentmanager cm
100244  symon
100245  pld genesil
100246  ctid
  cluster_transport_interface
100247  ccd
  cluster_configuration_db
100248  pmfd
100249  dmi2_client
100250  mfs_admin
100251  ndshared_unlink
100252  ndshared_touch
100253  ndshared_slink
100254  cbs control_board_server
100255  skiserv
100256  nfsxa nfsxattr
100257  ndshared_disable
100258  ndshared_enable
100259  sms_account_admin
100260  sms_modem_admin
100261  sms_r_login
100262  sms_r_subaccount_mgmt
100263  sms_service_admin
100264  session_admin
100265  canci_ancs_program
100266  canci_sms_program
100267  msmp
100268  halck
100269  halogmsg
100270  nfs_id_map
100271  ncall
100272  hmip
100273  repl_mig
100274  repl_mig_cb
<table>
<thead>
<tr>
<th>Application/Service</th>
<th>Port Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIS+</td>
<td>100300 - 100301</td>
<td>nisplus</td>
</tr>
<tr>
<td>NIS+</td>
<td>100302 - 100303</td>
<td>nis_cachemgr</td>
</tr>
<tr>
<td>NIS+ call back protocol</td>
<td>100304 - 100305</td>
<td>[unknown]</td>
</tr>
<tr>
<td>NIS+ Password Update Daemon</td>
<td>100306 - 100307</td>
<td>nispasswdd</td>
</tr>
<tr>
<td>FNS context update in NIS</td>
<td>100308 - 100309</td>
<td>[unknown]</td>
</tr>
<tr>
<td>unassigned</td>
<td>100310 - 100398</td>
<td></td>
</tr>
<tr>
<td>nfscksum</td>
<td>100399</td>
<td>nfscksum</td>
</tr>
<tr>
<td>network utilization agent</td>
<td>100400 - 100401</td>
<td>netmgmt_netu_progs</td>
</tr>
<tr>
<td>network rpc ping agent</td>
<td>100402 - 100403</td>
<td>na.shell</td>
</tr>
<tr>
<td>picsprint</td>
<td>100404</td>
<td>traps</td>
</tr>
<tr>
<td></td>
<td>100405 - 100409</td>
<td>[unknown]</td>
</tr>
<tr>
<td>rdmacconfig</td>
<td>100417</td>
<td>rpc.rdmaconfig</td>
</tr>
<tr>
<td>IETF NFSv4 Working Group - FedFS</td>
<td>100418 - 100421</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100422</td>
<td>mdcommd</td>
</tr>
<tr>
<td></td>
<td>100423</td>
<td>kiprop krb5_iprop</td>
</tr>
<tr>
<td></td>
<td>100424</td>
<td>stsf</td>
</tr>
<tr>
<td>unassigned</td>
<td>100425 - 100499</td>
<td></td>
</tr>
<tr>
<td>Sun Microsystems</td>
<td>100500 - 100531</td>
<td>[unknown]</td>
</tr>
<tr>
<td></td>
<td>100532</td>
<td>ucmmstate</td>
</tr>
<tr>
<td></td>
<td>100533</td>
<td>scrcmd</td>
</tr>
<tr>
<td>unassigned</td>
<td>100534 - 100999</td>
<td></td>
</tr>
<tr>
<td>nse link daemon</td>
<td>101002</td>
<td>nselinktool</td>
</tr>
<tr>
<td>nse link application</td>
<td>101003 - 101004</td>
<td>nselinkapp</td>
</tr>
<tr>
<td>unassigned</td>
<td>101005 - 101999</td>
<td></td>
</tr>
<tr>
<td>unassigned</td>
<td>101900 - 101901</td>
<td>[unknown]</td>
</tr>
<tr>
<td>unassigned</td>
<td>101902 - 101999</td>
<td></td>
</tr>
<tr>
<td>AssetLite</td>
<td>102000</td>
<td>[unknown]</td>
</tr>
<tr>
<td>PagerTool</td>
<td>102001</td>
<td>[unknown]</td>
</tr>
<tr>
<td>Discover</td>
<td>102002</td>
<td>[unknown]</td>
</tr>
<tr>
<td>unassigned</td>
<td>102003 - 105000</td>
<td></td>
</tr>
<tr>
<td>ShowMe</td>
<td>105001</td>
<td>sharedapp</td>
</tr>
<tr>
<td>Registry</td>
<td>105002</td>
<td>REGISTRY_PROG</td>
</tr>
<tr>
<td>Print-server</td>
<td>105003</td>
<td>print-server</td>
</tr>
<tr>
<td>Proto-server</td>
<td>105004</td>
<td>proto-server</td>
</tr>
<tr>
<td>Service/Category</td>
<td>Port-Range</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>----------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Notification-server</td>
<td>105005</td>
<td>notification-server</td>
</tr>
<tr>
<td>Transfer-agent-server</td>
<td>105006</td>
<td>transfer-agent-server</td>
</tr>
<tr>
<td>unassigned</td>
<td>105007 - 110000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>110001</td>
<td>tsolrpccb</td>
</tr>
<tr>
<td></td>
<td>110002</td>
<td>tsolpeerinfo</td>
</tr>
<tr>
<td></td>
<td>110003</td>
<td>tsolboot</td>
</tr>
<tr>
<td></td>
<td>120001</td>
<td>cmip na.cmip</td>
</tr>
<tr>
<td></td>
<td>120002</td>
<td>na.osidiscv</td>
</tr>
<tr>
<td></td>
<td>120003</td>
<td>cmiptrap</td>
</tr>
<tr>
<td>unassigned</td>
<td>120004 - 120099</td>
<td></td>
</tr>
<tr>
<td></td>
<td>120100</td>
<td>eserver</td>
</tr>
<tr>
<td></td>
<td>120101</td>
<td>repserver</td>
</tr>
<tr>
<td></td>
<td>120102</td>
<td>swserver</td>
</tr>
<tr>
<td></td>
<td>120103</td>
<td>dmd</td>
</tr>
<tr>
<td></td>
<td>120104</td>
<td>ca</td>
</tr>
<tr>
<td>unassigned</td>
<td>120105 - 120125</td>
<td></td>
</tr>
<tr>
<td></td>
<td>120126</td>
<td>nf_fddi</td>
</tr>
<tr>
<td></td>
<td>120127</td>
<td>nf_fddismt7_2</td>
</tr>
<tr>
<td>unassigned</td>
<td>120128 - 150000</td>
<td></td>
</tr>
<tr>
<td>pc passwd authorization</td>
<td>150001</td>
<td>pcnfsdprog</td>
</tr>
<tr>
<td>TOPS name mapping</td>
<td>150002</td>
<td>[unknown]</td>
</tr>
<tr>
<td>TOPS external attribute storage</td>
<td>150003</td>
<td>[unknown]</td>
</tr>
<tr>
<td>TOPS hierarchical file system</td>
<td>150004</td>
<td>[unknown]</td>
</tr>
<tr>
<td>TOPS NFS transparency extensions</td>
<td>150005</td>
<td>[unknown]</td>
</tr>
<tr>
<td>PC NFS License</td>
<td>150006</td>
<td>pcnfslicense</td>
</tr>
<tr>
<td>RDA</td>
<td>150007</td>
<td>rdaprog</td>
</tr>
<tr>
<td>WabiServer</td>
<td>150008</td>
<td>wsprog</td>
</tr>
<tr>
<td>WabiServer</td>
<td>150009</td>
<td>wsrlprog</td>
</tr>
<tr>
<td>unassigned</td>
<td>150010 - 160000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>160001</td>
<td>nihon-cm</td>
</tr>
<tr>
<td></td>
<td>160002</td>
<td>nihon-ce</td>
</tr>
<tr>
<td>unassigned</td>
<td>160003 - 170099</td>
<td></td>
</tr>
<tr>
<td></td>
<td>170100</td>
<td>domf_daemon0</td>
</tr>
<tr>
<td></td>
<td>170101</td>
<td>domf_daemon1</td>
</tr>
<tr>
<td></td>
<td>170102</td>
<td>domf_daemon2</td>
</tr>
<tr>
<td></td>
<td>170103</td>
<td>domf_daemon3</td>
</tr>
<tr>
<td></td>
<td>170104</td>
<td>domf_daemon4</td>
</tr>
<tr>
<td></td>
<td>170105</td>
<td>domf_daemon5</td>
</tr>
<tr>
<td>unassigned</td>
<td>170106 - 179999</td>
<td></td>
</tr>
<tr>
<td></td>
<td>180000</td>
<td>cecprog</td>
</tr>
<tr>
<td></td>
<td>180001</td>
<td>cecsysprog</td>
</tr>
<tr>
<td></td>
<td>180002</td>
<td>cec2cecprog</td>
</tr>
<tr>
<td></td>
<td>180003</td>
<td>cesprog</td>
</tr>
<tr>
<td></td>
<td>180004</td>
<td>ces2cesprog</td>
</tr>
<tr>
<td></td>
<td>180005</td>
<td>cet2cetprog</td>
</tr>
<tr>
<td></td>
<td>180006</td>
<td>cet2cetdoneprog</td>
</tr>
<tr>
<td></td>
<td>180007</td>
<td>cetcomprog</td>
</tr>
<tr>
<td></td>
<td>180008</td>
<td>cetsysprog</td>
</tr>
</tbody>
</table>
180009  cghapresenceprog
180010  cgdm-syncprog
180011  cgdmcnsccliprog
180012  cgdmrncsscliprog
180013  cgdmrncsssvcproG
180014  chmprg
180015  chmsysprog
180016  crcsapiproG
180017  ckptmprg
180018  crimcomponentprog
180019  crimqueryprog
180020  crimsecondaryprog
180021  crimservicesprog
180022  crimsyscomponentprog
180023  crimsysservicesprog
180024  csmagtapiproG
180025  csmagtcallbackprog
180026  csmreplicaprog
180027  csmrsvrprog
180028  csscltiprog
180029  csssvrprog
180030  csscopolstprog
180031 - 199999  unassigned
200000  pyramid_nfs
200001  pyramid_reserved
200002  cadds_image
200003  stellar_name_prog
200004  [unknown]
200005  [unknown]
200006  pacl
200007  lookupids
200008  ax_statd_prog
200009  ax_statd2_prog
200010  edm
200011  dtedirwd
200012  [unknown]
200013  [unknown]
200014  [unknown]
200015  [unknown]
200016  easerpcd
200017  rlxnfs
200018  sascuidprog
200019  knfsd
200020  ftncsd ftncsd_program
200021  ftsyncd ftsyncd_program
200022  ftstattd ftstattd_program
200023  exportmap
200024  nfs_metadata
unassigned
200025 - 200200
200201  ecoal
200202  eamon
200203  ecolic
200204  cs_printstatus_svr
200205  ecodisc

unassigned
200206 - 300000
300001  adt_rflockprog
300002  columbine
300003  system33_prog
300004  frame_prog1
300005  uimxprog
300006  rvd
300007  entombing daemon
300008  account mgmt system
300009  frame_prog2
300010  beeper access
300011  dptuprog
300012  mx-bcp
300013  instrument-file-access
300014  file-system-statistics
300015  unify-database-server
300016  tmd_msg
300017  [unknown]
300018  [unknown]
300019  automounter access
300020  lock server
300021  [unknown]
300022  office-automation-1
300023  office-automation-2
300024  office-automation-3
300025  office-automation-4
300026  office-automation-5
300027  office-automation-6
300028  office-automation-7
300029  local-data-manager
300030  chide
300031  csi_program
300032  [unknown]
300033  online-help
300034  case-tool
300035  delta
300036  rgi
300037  instrument-config-server
300038  [unknown]
300039  [unknown]
300040  dtia-rpc-server
300041  cms
viewer
aqm
exclaim
masterplan
fig_tool
[unknown]
[unknown]
[unknown]
remote-lock-manager
[unknown]
gdebug
ldebug
rscanner
[unknown]
[unknown]
[unknown]
snSERVER
[unknown]
[unknown]
[unknown]
[unknown]
[unknown]
[unknown]
[unknown]
BioStation
[unknown]
NetProb
Logging
Logging
[unknown]
[unknown]
[unknown]
[unknown]
[unknown]
[unknown]
sw_twin
remote_get_login
odcprog
[unknown]
[unknown]
[unknown]
[unknown]
300090 [unknown]
300091 smartdoc
300092 superping
300093 distributed-chembench
300094 uacman/alfil-uacman
300095 ait_rcagent_prog
300096 ait_rcagent_appl_prog
300097 smart
300098 ecoprog
300099 leonardo
300100 [unknown]
300101 [unknown]
300102 [unknown]
300103 [unknown]
300104 [unknown]
300105 [unknown]
300106 [unknown]
300107 [unknown]
300108 wingz
300109 teidan
300110 [unknown]
300111 [unknown]
300112 [unknown]
300113 [unknown]
300114 [unknown]
300115 [unknown]
300116 cadc_fhlockprog
300117 highscan
300118 [unknown]
300119 [unknown]
300120 [unknown]
300121 opennavigator
300122 aarpcxfer
300123 [unknown]
300124 [unknown]
300125 [unknown]
300126 groggs
300127 licsrv
300128 issdemon
300129 [unknown]
300130 maximize
300131 cgms_server
300132 [unknown]
300133 agent_rpc
300134 docmaker
300135 docmaker
300136 [unknown]
300137 [unknown]
300138  [unknown]
300139  iesx
300140  [unknown]
300141  [unknown]
300142  [unknown]
300143  [unknown]
300144  smart-mbs
300145  [unknown]
300146  [unknown]
300147  docimage
300148  [unknown]
300149  dmc-interface
300150  [unknown]
300151  jss
300152  [unknown]
300153  arimage
300154  xdb-workbench
300155  frontdesk
300156  dmc
300157  expressight-6000
300158  graph service program
300159  [unknown]
300160  [unknown]
300161  [unknown]
300162  [unknown]
300163  [unknown]
300164  [unknown]
300165  [unknown]
300166  [unknown]
300167  [unknown]
300168  [unknown]
300169  [unknown]
300170  [unknown]
300171  [unknown]
300172  [unknown]
300173  [unknown]
300174  [unknown]
300175  [unknown]
300176  rlpr
300177  nx_hostdprog
300178  netuser-x
300179  rmntprog
300180  [unknown]
300181  mipe
300182  [unknown]
300183  collectorprog
300184  uslookup_PROG
300185  viewstation
iate
[unknown]
[unknown]
imsvtprog
[unknown]
[unknown]
[unknown]
[unknown]
pmdb
pmda
[unknown]
[unknown]
trend_idbd
rres
sd.masterd
sd.executiond
sd.listend
sd.reserve1
sd.reserve2
msbd
stagedprog
mountprog
watchdprog
pms
[unknown]
session_server_program
session_program
debug_serverprog
[unknown]
[unknown]
paceprog
[unknown]
mbus
aframes2ps
npartprog
cmlserver
cmlbridge
sailfrogfaxprog
sailfrogphoneprog
sailfrogvmailprog
wserviceprog arcstorm
hdl
alive
radsp
radavx
radview
rays_prog
rays_prog
<table>
<thead>
<tr>
<th>Program Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>fm_rpc_prog</td>
</tr>
<tr>
<td>aries</td>
</tr>
<tr>
<td>uapman</td>
</tr>
<tr>
<td>ddman</td>
</tr>
<tr>
<td>top</td>
</tr>
<tr>
<td>[unknown]</td>
</tr>
<tr>
<td>trendlink</td>
</tr>
<tr>
<td>licenseprog</td>
</tr>
<tr>
<td>statuslicenseprog</td>
</tr>
<tr>
<td>oema_rmpf_svc</td>
</tr>
<tr>
<td>oema_smpf_svc</td>
</tr>
<tr>
<td>oema_rmsg_svc</td>
</tr>
<tr>
<td>grapes-sd</td>
</tr>
<tr>
<td>ds_master</td>
</tr>
<tr>
<td>ds_transfer</td>
</tr>
<tr>
<td>ds_logger</td>
</tr>
<tr>
<td>ds_query</td>
</tr>
<tr>
<td>[unknown]</td>
</tr>
<tr>
<td>[unknown]</td>
</tr>
<tr>
<td>nsd_prog</td>
</tr>
<tr>
<td>browser</td>
</tr>
<tr>
<td>epoch</td>
</tr>
<tr>
<td>floorplanner</td>
</tr>
<tr>
<td>reach</td>
</tr>
<tr>
<td>tactic</td>
</tr>
<tr>
<td>cachescientific1</td>
</tr>
<tr>
<td>cachescientific2</td>
</tr>
<tr>
<td>desksrc_prog</td>
</tr>
<tr>
<td>photo3d1</td>
</tr>
<tr>
<td>photo3d2</td>
</tr>
<tr>
<td>[unknown]</td>
</tr>
<tr>
<td>soundmgr</td>
</tr>
<tr>
<td>s6k</td>
</tr>
<tr>
<td>aims_referenced_text_processor</td>
</tr>
<tr>
<td>xess</td>
</tr>
<tr>
<td>ds_queue</td>
</tr>
<tr>
<td>[unknown]</td>
</tr>
<tr>
<td>orionscanplus</td>
</tr>
<tr>
<td>openlink-xx</td>
</tr>
<tr>
<td>kbmsprog</td>
</tr>
<tr>
<td>[unknown]</td>
</tr>
<tr>
<td>futuresource</td>
</tr>
<tr>
<td>the_xprt</td>
</tr>
<tr>
<td>cmg_srvprog</td>
</tr>
<tr>
<td>[unknown]</td>
</tr>
<tr>
<td>[unknown]</td>
</tr>
<tr>
<td>front</td>
</tr>
</tbody>
</table>
300281  [unknown]
300282  [unknown]
300283  [unknown]
300284  conmanprog
300285  jincv2
300286  lsIs
300287  systemstatprog
300288  fxpsprog
300289  callpath
300290  axess
300291  armor_rpcd
300292  armor_dictionary_rpcd
300293  armor_miscd
300294  filetransfer_prog
300295  bl_swda
300296  bl_hwda
300297  [unknown]
300298  [unknown]
300299  [unknown]
300300  filemon
300301  acunetprog
300302  rbuild
300303  assistprog
300304  tog
300305  [unknown]
300306  sns7000
300307  igprog
300308  tgprog
300309  plc
300310  pxman pxlsprog
300311  hde_server hdeserver
300312  tsslicenseprog
300313  rpc.explorerd
300314  chrld
300315  tbisam
300316  tbis
300317  adsprog
300318  sponsorprog
300319  querycmprog
300320  [unknown]
300321  [unknown]
300322  mobill
300323  sld
   service_locator_daemon
300324  linkprog
300325  codexdaemonprog
300326  drprog
300327  ressys_commands
<p>| 300328 | stamp          |
| 300329 | matlab         |
| 300330 | sched1d        |
| 300331 | upcprog        |
| 300332 | xferbkch       |
| 300333 | xfer           |
| 300334 | qbthd          |
| 300335 | qbabort        |
| 300336 | lsd            |
| 300337 | geomgrd        |
| 300338 | generic_fts    |
| 300339 | ft_ack         |
| 300340 | lymb           |
| 300341 | vantage        |
| 300342 | cltstd         |
| 300343 | clooptstdprog   |
| 300344 | clui_clui_prog |
| 300345 | testerd        |
| 300346 | tstdprog       |
| 300347 | extsim         |
| 300348 | cmd_dispatch   |
| 300349 | maxm_ems       |
| 300350 | callpath_receive_program |
| 300351 | x3270prog      |
| 300352 | sbc_lag        |
| 300353 | sbc_frsa       |
| 300354 | sbc_frs        |
| 300355 | atommgr        |
| 300356 | geostrat       |
| 300357 | dbvialu6.2     |
| 300358 | [unknown]      |
| 300359 | [unknown]      |
| 300360 | [unknown]      |
| 300361 | [unknown]      |
| 300362 | clarion mgr    |
| 300363 | setcimrpc      |
| 300364 | virtual_protocol_adapter |
| 300365 | unibart        |
| 300366 | uniarch        |
| 300367 | unifile        |
| 300368 | unisrex        |
| 300369 | uniscmd        |
| 300370 | rsc            |
| 300371 | set            |
| 300372 | desaf-ws/key   |
| 300373 | reeldb         |
| 300374 | nl             |
| 300375 | rmd            |</p>
<table>
<thead>
<tr>
<th>Process ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>300376</td>
<td>agcd</td>
</tr>
<tr>
<td>300377</td>
<td>rsynd</td>
</tr>
<tr>
<td>300378</td>
<td>rcnlib</td>
</tr>
<tr>
<td>300379</td>
<td>rcnlib_attach</td>
</tr>
<tr>
<td>300380</td>
<td>evergreen_mgnt_agent</td>
</tr>
<tr>
<td>300381</td>
<td>fx104prog</td>
</tr>
<tr>
<td>300382</td>
<td>rui</td>
</tr>
<tr>
<td>300383</td>
<td>remote_user_interface</td>
</tr>
<tr>
<td>300384</td>
<td>ovomd</td>
</tr>
<tr>
<td>300385</td>
<td>[unknown]</td>
</tr>
<tr>
<td>300386</td>
<td>system_server</td>
</tr>
<tr>
<td>300387</td>
<td>pipecs cs_pipeprog</td>
</tr>
<tr>
<td></td>
<td>ppktrpc</td>
</tr>
<tr>
<td>300388</td>
<td>uv-net univision</td>
</tr>
<tr>
<td>300389</td>
<td>auexe</td>
</tr>
<tr>
<td>300390</td>
<td>audip</td>
</tr>
<tr>
<td>300391</td>
<td>mqi</td>
</tr>
<tr>
<td>300392</td>
<td>eva</td>
</tr>
<tr>
<td>300393</td>
<td>eeei_reserved_1</td>
</tr>
<tr>
<td>300394</td>
<td>eeei_reserved_2</td>
</tr>
<tr>
<td>300395</td>
<td>eeei_reserved_3</td>
</tr>
<tr>
<td>300396</td>
<td>eeei_reserved_4</td>
</tr>
<tr>
<td>300397</td>
<td>eeei_reserved_5</td>
</tr>
<tr>
<td>300398</td>
<td>eeei_reserved_6</td>
</tr>
<tr>
<td>300399</td>
<td>eeei_reserved_7</td>
</tr>
<tr>
<td>300400</td>
<td>eeei_reserved_8</td>
</tr>
<tr>
<td>300401</td>
<td>cprlm</td>
</tr>
<tr>
<td>300402</td>
<td>wg_idms_manager</td>
</tr>
<tr>
<td>300403</td>
<td>timequota</td>
</tr>
<tr>
<td>300404</td>
<td>spiff</td>
</tr>
<tr>
<td>300405-300414</td>
<td>ov_oem_svc</td>
</tr>
<tr>
<td>300415</td>
<td>ov_msg_ctlg_svc</td>
</tr>
<tr>
<td>300416</td>
<td>ov_advt_reg_svc</td>
</tr>
<tr>
<td>300417-300424</td>
<td>showkron</td>
</tr>
<tr>
<td>300425</td>
<td>daatd</td>
</tr>
<tr>
<td>300426</td>
<td>swiftnet</td>
</tr>
<tr>
<td>300427</td>
<td>ovomdel</td>
</tr>
<tr>
<td>300428</td>
<td>ovomreq</td>
</tr>
<tr>
<td>300429</td>
<td>msg_dispatcher</td>
</tr>
<tr>
<td>300430</td>
<td>pcshare server</td>
</tr>
<tr>
<td>300431</td>
<td>rcvs</td>
</tr>
<tr>
<td>300432</td>
<td>fdfsver</td>
</tr>
<tr>
<td>300433</td>
<td>bssd</td>
</tr>
<tr>
<td>300434</td>
<td>drdd</td>
</tr>
<tr>
<td>300435</td>
<td>mif_gutsprog</td>
</tr>
<tr>
<td>300436</td>
<td>mif_guiprog</td>
</tr>
<tr>
<td>300437</td>
<td>twolfd</td>
</tr>
</tbody>
</table>
300438  twscd
300439  nwsbumv
300440  dgux_mgr
300441  pfxd
300442  tds
300443  ovomadmind
300444  ovomgate
300445  omadmind
300446  nps
300447  npd
300448  tsa
300449  cdaimc

unassigned
300450-300452
300453  ckt_implementation
300454  mda-tactical

unassigned
300455-300458
300459  atrrun
300460  RoadRunner
300461  nas
300462  undelete
300463  ovacadd
300464  tbdesmai
300465  arguslm
300466  dmd
300467  drd
300468  fm_help
300469  ftransrpc_prog
300470  finrisk
300471  dg_pc_idisched
300472  dg_pc_idiserv
300473  apd
300474  ap_sspd
300475  callpatheventrecorder
300476  flc
300477  dg_osm
300478  dspnamed
300479  iqddsrv
300480  iqjobsrv
300481  tacosxx
300482  wheeldbmg
300483  cnxmgr_nm_prog
300484  cnxmgr_cfg_prog
300485  3dsmapper
300486  ids
300487  imagine_rpc_svc
300488  lfn
300489  salesnet
300490  defaxo
300491 dbqtsd
300492 kms
300493 rpc.iced
300494 calc2s
300495 ptouidprog
300496 docs1s
300497 new
300498 collagebdg
300499 ars_server
300500 ars_client
300501 vr_catalog
300502 vr_tdb
300503 ama
300504 evama
300505 conama
300506 service_process
300507 reuse_proxy
300508 mars_ctrl
300509 mars_db
300510 mars_com
300511 mars_admch
300512 tbpiccip
300513 tcp_acs_svc
300514 inout_svc
300515 csoft_wp
300516 mcfs
300517 eventprog
300518 dg_pc_idimsg
300519 dg_pc_idiaux
300520 atsr_gc
300521 alarm alarm_prog
300522 fts_prog
300523 dcs_prog
300524 ihb_prog
300525 [unknown]
300526 [unknown]
300527 clu_info_prog
300528 rmfm
300529 c2sdocd
300530 interahelp
300531 callpathasyncmsghandler
300532 optix_arc
300533 optix_ts
300534 optix_wf
300535 maxopenc
300536 cev cev_server
300537 sitewideprog
300538 drs
300539  drsdm
300540  dasgate
300541  dcdbd
300542  dcpssd
300543  supportlink_prog
300544  broker
300545  listener
300546  multiaccess
300547  spai_interface
300548  spai_adaption
300549  chimera_ci
   chimera_clientinterface
300550  chimera_pi
   chimera_processinvoker
300551  teamware_f1
   teamware_foundationlevel
300552  teamware_sl
   teamware_systemlevel
300553  teamware_ui
   teamware_userinterface
300554  lprm
300555  mpsprog
   Mensuration_Proxy_Server
300556  mo_symdis
300557  retsideprog
300558  slp
300559  slm-api
300560  im_rpc_teamconference
300561  license_prog_license
300562  stuple_stuple_prog
300563  upasswd_prog
300564  gentranmentorsecurity
300565  gentranmentorprovider
300566  latituded
   latitude_license_server
300567  gentranmentorreq1
300568  gentranmentorreq2
300569  gentranmentorreq3
300570  rj_server
300571  gws-rdb
300572  gws-mpmd
300573  gws-spmd
300574  vwcalcd
300575  vworad
300576  vwsybd
300577  vwave
300578  online_assistant
300579  internet_assistant
spawnd
procmgrg
cfgdbd
logutild
ibis
ibisaux
aapi
rstrt
hbeat
pcspu
empress
sched_server
LiveScheduler
path_server
LiveScheduler
c2sdmd
c2scf
btsas
dtas
appie
dmi
pscd
panther software corp daemon
sisd
cpwebserver
wwcommo
mx-mie
mx-mie-debug
idmn
ssrv
vpnservice
samserver
sams_server
chrysalis
ddm
ddm
mx-bcp-debug
upmrd
upmdsd
res
colortron
zrs
afpsrv
apxft
nrp
hpid
mailwatch
fos bc_fcrb_receiver
300625    cs_sysadmin_svr
300626    cs_controller_svr
300627    nokia_nms_eai
300628    dbg
300629    remex
300630    cs_bind
300631    idm
300632    prpasswd
300633    iw-pw
300634    starrb
300635    Impress_Server
300636    colorstar
300637    gwugui
300638    gwsgui
300639    dai_command_proxy
300640    dai_alarm_server
300641    dai_fui_proxy
300642    spai_command_proxy
300643    spai_alarm_server
300644    iris
300645    hcxttp
300646    updatedb rsched
300647    urnd urn
300648    iqwpsrv
300649    dskutild
300650    online
300651    nlserv
300652    acsm
300653    dg_clar_sormsg
300654    wwpollerrpc
300655    wwwmodelrpc
300656    nsprofd
300657    nsdistd
300658    recollect
300659    lsexecd lss_res
300660    lssagend lss_rea
300661    cdinfo
300662    sninsr_addon
300663    mm-sap
300664    ks
300665    psched
300666    tekdvfs
300667    storxll
300668    nisse
300669    lbadvise
300670    atcinstaller
300671    atntstarter
300672    NetML
300673  tdmesmge
300674  tdmesmgd
300675  tdmesmgt
300676  oim
300677  mediamanagement
300678  rdbprog fieldowsrv
300679  rpwdprogrpwd
300680  sapi-trace
300681  sapi-master-daemon
300682  omdcuprog om-dcu
300683  wwpromon
300684  tndidprog
300685  rkey_setsecretprog
300686  asdu_server_progrddd
300687  pwrctnl
300688  siunixd
300689  wmapi
300690  cross_reference_ole
300691  rtc
300692  disp
300693  sql_compilation_agent
300694  tnsysprogrddd
300695  ius-sapimd
300696  apteam-dxrddd
300697  rmsrpc
300698  seismic_system
300699  remote
300700  ttl_ts_event nokia_nms
300701  fxrr
300702  onlicense
300703  vxkey
300704  dinis
300705  sched2d schedule-2
300706  sched3d schedule-3
300707  sched4d schedule-4
300708  sched5d schedule-5
300709  sched6d schedule-6
300710  sched7d schedule-7
300711  sched8d schedule-8
300712  sched9d schedule-9
300713  adtsqry
300714  adserv
300715  adrepserv
300716  [unknown]
300717  caad
300718  caau
300719  cescda
300720  vcapiaadmin

Thurlow                     Standards Track                    [Page 49]
300721  vcapi20
300722  tcfs
300723  csed
300724  nothand
300725  hacb
300726  nfauth
300727  imlm
300728  bestcomm
300729  lprpasswd
300730  rprpasswd
300731  proplistd
300732  mikomomc
300733  arepa-cas
300734  [unknown]
300735  [unknown]
300736  ando_ts
300737  intermezzo
300738  ftel-sdh-request
300739  ftel-sdh-response
300740  [unknown]
300741  [unknown]
300742  [unknown]
300743  [unknown]
300744  [unknown]
300745  vrc_abb
300746  vrc_comau
300747  vrc_fanuc
300748  vrc_kuka
300749  vrc_reis
300750  hp_sv6d
300751  correntmgr01
300752  correntike
300753  [unknown]
300754  [unknown]
300755  intransa_location
300756  intransa_management
300757  intransa_federation
300758  portprot
300759  ipmiprot
300760  aceapi
300761  f6000pss
300762  vsmapi_program
300763  ubertuple
300764  ctconcrpcif
300765  mfuadmin
300766  aiols
300767  dsmrootd
300768  htdl
| 300769  | cab 
| 300770  | vrc_cosimir 
| 300771  | cmhelmd 
| 300772  | polynsm 
| 300773  | [unknown] 
| 300774  | [unknown] 
| 300775  | [unknown] 
| 300776  | [unknown] 
| 300777  | [unknown] 
| 300778  | [unknown] 
| 300779  | [unknown] 
| 300780  | [unknown] 
| 300781  | dsmrecallld 
| 300782  | [unknown] 
| 300783  | [unknown] 
| 300784  | twrgcontrol 
| 300785  | twired 
| 300786  | twrcfgdb 

BMC software
300787-300886
unassigned
300887 - 300999

Sun Microsystems
301000-302000 [ 2000 numbers ]
unassigned
302001-349999

American Airlines
350000 - 350999

Acucobol Inc.
351000 - 351099

The Bristol Group
351100 - 351249

Amteva Technologies
351250 - 351349

351350  | wfmMgmtApp 
351351  | wfmMgmtDataSrv 
351352  | wfmMgmtFut1 
351353  | wfmMgmtFut1 
351354  | wfmAPM 
351355  | wfmIAMgr 
351356  | wfmECMgr 
351357  | wfmLookOut 
351358  | wfmAgentFut1 
351359  | wfmAgentFut2 

unassigned
351360 - 351406

Sterling Software ITD
351407  | csed 
351360  | sched10d 
351361  | sched11d 
351362  | sched12d 
351363  | sched13d 
351364  | sched14d 
351365  | sched15d 
351366  | sched16d 
351367  | sched17d 
351368  | sched18d 
351369  | sched19d 

Thurlow Standards Track [Page 51]
351370 sched20d
351371 sched21d
351372 sched22d
351373 sched23d
351374 sched24d
351375 sched25d
351376 sched26d
351377 sched27d
351378 sched28d
351379 sched29d
351380 sched30d
351381 sched31d
351382 sched32d
351383 sched33d
351384 sched34d
351385 sched35d
351386 sched36d
351387 sched37d
351388 sched38d
351389 sched39d
351390 consoleserver
351391 scheduleserver
351392 RDELIVER
351393 REVENTPROG
351394 RSENDEVENTPROG
351395 snapp
351396 snapad
351397 sdsoodb
351398 sdsmain
351399 sdssrv
351400 sdscow
351401 sdsserv
351402 fsbatch
351403 fsmonitor
351404 fsdisp
351405 fssession
351406 fslog
351407 svdpappserv
351408 gns
351409 [unknown]
351410 [unknown]
351411 [unknown]
351412 axi
351413 rpcxfr
351414 slm
351415 smbpasswd
351416 tbdbserv
351417 tbprojserv
<table>
<thead>
<tr>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>351418</td>
<td>genericserver</td>
</tr>
<tr>
<td>351419</td>
<td>dynarc_ds</td>
</tr>
<tr>
<td>351420</td>
<td>dnscmdr</td>
</tr>
<tr>
<td>351421</td>
<td>ipcmdr</td>
</tr>
<tr>
<td>351422</td>
<td>faild</td>
</tr>
<tr>
<td>351423</td>
<td>failmon</td>
</tr>
<tr>
<td>351424</td>
<td>faildebug</td>
</tr>
<tr>
<td>351425</td>
<td>[unknown]</td>
</tr>
<tr>
<td>351426</td>
<td>[unknown]</td>
</tr>
<tr>
<td>351427</td>
<td>siemens_srs</td>
</tr>
<tr>
<td>351428</td>
<td>bsproxy</td>
</tr>
<tr>
<td>351429</td>
<td>ifsrpc</td>
</tr>
<tr>
<td>351430</td>
<td>CesPvcSm</td>
</tr>
<tr>
<td>351431</td>
<td>FrPvcSm</td>
</tr>
<tr>
<td>351432</td>
<td>AtmPvcSm</td>
</tr>
<tr>
<td>351433</td>
<td>radius</td>
</tr>
<tr>
<td>351434</td>
<td>auditor</td>
</tr>
<tr>
<td>351435</td>
<td>sft</td>
</tr>
<tr>
<td>351436</td>
<td>voicemail</td>
</tr>
<tr>
<td>351437</td>
<td>kis</td>
</tr>
<tr>
<td>351438</td>
<td>SOFTSERV_NOTIFY</td>
</tr>
<tr>
<td>351439</td>
<td>dynarpc</td>
</tr>
<tr>
<td>351440</td>
<td>hc</td>
</tr>
<tr>
<td>351441</td>
<td>iopas</td>
</tr>
<tr>
<td>351442</td>
<td>iopcs</td>
</tr>
<tr>
<td>351443</td>
<td>iopss</td>
</tr>
<tr>
<td>351444</td>
<td>spcnfs</td>
</tr>
<tr>
<td>351445</td>
<td>spcvss</td>
</tr>
<tr>
<td>351446</td>
<td>matilda_sms</td>
</tr>
<tr>
<td>351447</td>
<td>matilda_brs</td>
</tr>
<tr>
<td>351448</td>
<td>matilda_dbs</td>
</tr>
<tr>
<td>351449</td>
<td>matilda_sps</td>
</tr>
<tr>
<td>351450</td>
<td>matilda_svs</td>
</tr>
<tr>
<td>351451</td>
<td>matilda_sds</td>
</tr>
<tr>
<td>351452</td>
<td>matilda_vvs</td>
</tr>
<tr>
<td>351453</td>
<td>matilda_stats</td>
</tr>
<tr>
<td>351454</td>
<td>xtrade</td>
</tr>
<tr>
<td>351455</td>
<td>mapsvr</td>
</tr>
<tr>
<td>351456</td>
<td>hp_graphicsd</td>
</tr>
<tr>
<td>351457</td>
<td>berkeley_db</td>
</tr>
<tr>
<td></td>
<td>berkeley_db_svc</td>
</tr>
<tr>
<td>351458</td>
<td>io_server</td>
</tr>
<tr>
<td>351459</td>
<td>rpc.niod</td>
</tr>
<tr>
<td>351460</td>
<td>rpc.kill</td>
</tr>
<tr>
<td>351461</td>
<td>hmdisproxy</td>
</tr>
<tr>
<td>351462</td>
<td>smdisproxy</td>
</tr>
<tr>
<td>351463</td>
<td>avatard</td>
</tr>
<tr>
<td>351464</td>
<td>namu</td>
</tr>
</tbody>
</table>
351465  BMCSess
351466  FENS_Sport
351467  EMCONFIG
351468  EMCONFIG_RESP
351469  lodge_proo
351470  ARCserveIT-Queue
351471  ARCserveIT-Device
351472  ARCserveIT-Discover
351473  ARCserveIT-Alert
351474  ARCserveIT-Database
351475  scand1
351476  scand2
351477  scand3
351478  scand4
351479  scand5
351480  dscv
351481  cb_svc
351482  [unknown]
351483  iprobe
351484  omniconf
351485  isan

BG Partners
351486 - 351500
351501  mond
351502  iqlremote
351503  iqlalarm

unassigned
351504 - 351599

Orion Multisystems
351600-351855

unassigned
351856 - 351899

NSP lab
351900 - 351999

unassigned
351999 - 352232

352233  asautostart
352234  asmediad1
352235  asmediad2
352236  asmediad3
352237  asmediad4
352238  asmediad5
352239  asmediad6
352240  asmediad7
352241  asmediad8
352242  asmediad9
352243  asmediad10
352244  asmediad11
352245  asmediad12
352246  asmediad13
352247  asmediad14
352248  asmediad15
352249  asmediad16
352250  waruser
<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>352251</td>
<td>warlogd</td>
</tr>
<tr>
<td>352252</td>
<td>warsvrmgr</td>
</tr>
<tr>
<td>352253</td>
<td>warvfaysd</td>
</tr>
<tr>
<td>352254</td>
<td>warftpd</td>
</tr>
<tr>
<td>352255</td>
<td>warnfds</td>
</tr>
<tr>
<td>352256</td>
<td>bofproxyc0</td>
</tr>
<tr>
<td>352257</td>
<td>bofproxys0</td>
</tr>
<tr>
<td>352258</td>
<td>bofproxyc1</td>
</tr>
<tr>
<td>352259</td>
<td>bofproxys1</td>
</tr>
<tr>
<td>352260</td>
<td>bofproxyc2</td>
</tr>
<tr>
<td>352261</td>
<td>bofproxys2</td>
</tr>
<tr>
<td>352262</td>
<td>bofproxyc3</td>
</tr>
<tr>
<td>352263</td>
<td>bofproxys3</td>
</tr>
<tr>
<td>352264</td>
<td>bofproxys4</td>
</tr>
<tr>
<td>352265</td>
<td>bofproxys5</td>
</tr>
<tr>
<td>352266</td>
<td>bofproxyc5</td>
</tr>
<tr>
<td>352267</td>
<td>bofproxys6</td>
</tr>
<tr>
<td>352268</td>
<td>bofproxyc6</td>
</tr>
<tr>
<td>352269</td>
<td>bofproxys7</td>
</tr>
<tr>
<td>352270</td>
<td>bofproxyc7</td>
</tr>
<tr>
<td>352271</td>
<td>bofproxys7</td>
</tr>
<tr>
<td>352272</td>
<td>bofproxyc8</td>
</tr>
<tr>
<td>352273</td>
<td>bofproxys8</td>
</tr>
<tr>
<td>352274</td>
<td>bofproxyc9</td>
</tr>
<tr>
<td>352275</td>
<td>bofproxys9</td>
</tr>
<tr>
<td>352276</td>
<td>bofproxyc9</td>
</tr>
<tr>
<td>352277</td>
<td>bofproxysa</td>
</tr>
<tr>
<td>352278</td>
<td>bofproxycb</td>
</tr>
<tr>
<td>352279</td>
<td>bofproxysb</td>
</tr>
<tr>
<td>352280</td>
<td>bofproxycb</td>
</tr>
<tr>
<td>352281</td>
<td>bofproxysc</td>
</tr>
<tr>
<td>352282</td>
<td>bofproxycd</td>
</tr>
<tr>
<td>352283</td>
<td>bofproxysd</td>
</tr>
<tr>
<td>352284</td>
<td>bofproxycd</td>
</tr>
<tr>
<td>352285</td>
<td>bofproxysd</td>
</tr>
<tr>
<td>352286</td>
<td>bofproxycf</td>
</tr>
<tr>
<td>352287</td>
<td>bofproxysf</td>
</tr>
<tr>
<td>352288</td>
<td>bofproxypo0</td>
</tr>
<tr>
<td>352289</td>
<td>bofproxypo1</td>
</tr>
<tr>
<td>352290</td>
<td>bofproxypo2</td>
</tr>
<tr>
<td>352291</td>
<td>bofproxypo3</td>
</tr>
<tr>
<td>352292</td>
<td>bofproxypo4</td>
</tr>
</tbody>
</table>

unassigned

352293–370000

<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>370001</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370002</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370003</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370004</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370005</td>
<td>[unknown]</td>
</tr>
<tr>
<td>Protocol Number</td>
<td>Service Name</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>370006</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370007</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370008</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370009</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370010</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370011</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370012</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370013</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370014</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370015</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370016</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370017</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370018</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370019</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370020</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370021</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370022</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370023</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370024</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370025</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370026</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370027</td>
<td>[unknown]</td>
</tr>
<tr>
<td>370028 - 379999</td>
<td>unassigned</td>
</tr>
<tr>
<td>380000</td>
<td>opensna</td>
</tr>
<tr>
<td>380001</td>
<td>probenet</td>
</tr>
<tr>
<td>380002</td>
<td>[unknown]</td>
</tr>
<tr>
<td>380003</td>
<td>license</td>
</tr>
<tr>
<td>380004</td>
<td>na.3com-remote</td>
</tr>
<tr>
<td>380005</td>
<td>na.ntp</td>
</tr>
<tr>
<td>380006</td>
<td>probeutil</td>
</tr>
<tr>
<td>380007</td>
<td>na.vlb</td>
</tr>
<tr>
<td>380008</td>
<td>cds_mhs_agent</td>
</tr>
<tr>
<td>380009</td>
<td>cds_x500_agent</td>
</tr>
<tr>
<td>380010</td>
<td>cds_mailhub_agent</td>
</tr>
<tr>
<td>380011</td>
<td>codex_6500_proxy</td>
</tr>
<tr>
<td>380012</td>
<td>codex_6500_trapd</td>
</tr>
<tr>
<td>380013</td>
<td>na.nm212</td>
</tr>
<tr>
<td>380014</td>
<td>cds_mta_metrics_agent</td>
</tr>
<tr>
<td>380015</td>
<td>[unknown]</td>
</tr>
<tr>
<td>380016</td>
<td>na.cable</td>
</tr>
<tr>
<td>380017</td>
<td>codexcapletrap</td>
</tr>
<tr>
<td>380018 - 380028</td>
<td>Swiss Re</td>
</tr>
<tr>
<td>380029</td>
<td>ncstat</td>
</tr>
<tr>
<td>380030</td>
<td>ncnfsstat</td>
</tr>
<tr>
<td>380031</td>
<td>ftams</td>
</tr>
<tr>
<td>380032</td>
<td>na.isotp</td>
</tr>
<tr>
<td>380033</td>
<td>na.rfc1006</td>
</tr>
<tr>
<td>380034 - 389999</td>
<td>unassigned</td>
</tr>
<tr>
<td>Organization</td>
<td>Numbers</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Epoch Systems</td>
<td>390000 - 390049</td>
</tr>
<tr>
<td>Quickturn Systems</td>
<td>390050 - 390065</td>
</tr>
<tr>
<td>Team One Systems</td>
<td>390066 - 390075</td>
</tr>
<tr>
<td>General Electric CRD</td>
<td>390076 - 390085</td>
</tr>
<tr>
<td>TSIG NFS subcommittee</td>
<td>390086 - 390089</td>
</tr>
<tr>
<td>SoftLab ab</td>
<td>390090 - 390099</td>
</tr>
<tr>
<td>Legato Network Services</td>
<td>390100 - 390115</td>
</tr>
<tr>
<td></td>
<td>390116</td>
</tr>
<tr>
<td></td>
<td>390117</td>
</tr>
<tr>
<td></td>
<td>390118</td>
</tr>
<tr>
<td></td>
<td>390119</td>
</tr>
<tr>
<td></td>
<td>390120</td>
</tr>
<tr>
<td></td>
<td>390121</td>
</tr>
<tr>
<td>Data General</td>
<td>390122 - 390141</td>
</tr>
<tr>
<td>Perfect Byte</td>
<td>390142 - 390171</td>
</tr>
<tr>
<td>JTS Computer Systems</td>
<td>390172 - 390181</td>
</tr>
<tr>
<td>Parametric Technology</td>
<td>390182 - 390191</td>
</tr>
<tr>
<td>Voxem</td>
<td>390192 - 390199</td>
</tr>
<tr>
<td>Effix Systems</td>
<td>390200 - 390299</td>
</tr>
<tr>
<td>Motorola</td>
<td>390300 - 390309</td>
</tr>
<tr>
<td>Mobile Data Intl.</td>
<td>390310 - 390325</td>
</tr>
<tr>
<td>Physikalisches Institut</td>
<td>390326 - 390330</td>
</tr>
<tr>
<td>Ergon Informatik AG</td>
<td>390331 - 390340</td>
</tr>
<tr>
<td>Analog Devices Inc.</td>
<td>390341 - 390348</td>
</tr>
<tr>
<td>Interphase Corporation</td>
<td>390349 - 390358</td>
</tr>
<tr>
<td>NeWsware</td>
<td>390359 - 390374</td>
</tr>
<tr>
<td>Qualix Group</td>
<td>390375 - 390379</td>
</tr>
<tr>
<td>Xerox Imaging Systems</td>
<td>390380 - 390389</td>
</tr>
<tr>
<td>Noble Net</td>
<td>390390 - 390399</td>
</tr>
<tr>
<td>Legato Network Services</td>
<td>390400 - 390499</td>
</tr>
<tr>
<td>Client Server Tech.</td>
<td>390500 - 390511</td>
</tr>
<tr>
<td>Atria</td>
<td>390512 - 390517</td>
</tr>
<tr>
<td>GE NMR Instruments</td>
<td>390518 - 390525</td>
</tr>
<tr>
<td>Harris Corp.</td>
<td>390526 - 390530</td>
</tr>
<tr>
<td>Unisys</td>
<td>390531 - 390562</td>
</tr>
<tr>
<td>Aggregate Computing</td>
<td>390563 - 390572</td>
</tr>
<tr>
<td>Interactive Data</td>
<td>390573 - 390580</td>
</tr>
<tr>
<td>OKG AB</td>
<td>390581 - 390589</td>
</tr>
<tr>
<td>K2 Software</td>
<td>390591 - 390594</td>
</tr>
<tr>
<td>Collier Jackson</td>
<td>390595 - 390599</td>
</tr>
<tr>
<td>Remedy Corporation</td>
<td>390600 - 390699</td>
</tr>
<tr>
<td>Mentor Graphics</td>
<td>390700 - 390799</td>
</tr>
<tr>
<td>AT&amp;T Bell Labs (Lucent)</td>
<td>390800 - 390899</td>
</tr>
<tr>
<td>Xerox</td>
<td>390900 - 390999</td>
</tr>
<tr>
<td>Silicon Graphics</td>
<td>391000 - 391063</td>
</tr>
<tr>
<td>Data General</td>
<td>391064 - 391095</td>
</tr>
<tr>
<td>Computer Support Corp.</td>
<td>391096 - 391099</td>
</tr>
<tr>
<td>Quorum Software Systems</td>
<td>391100 - 391199</td>
</tr>
</tbody>
</table>
InterLinear Technology 391200 – 391209
Highland Software 391210 – 391229
Boeing Comp. Svcs. 391230 – 391249
IBM Sweden 391250 – 391259
Signature Authority Svc 391260 – 391271
ZUMTOBEL Licht GmbH 391272 – 391283
NOAA/ERL 391284 – 391299
NCR Corp. 391300 – 391399
FTP Software 391400 – 391409
Cadre Technologies 391410 – 391433
Visionware Ltd (UK) 391434 – 391439
IBR-Partner AG 391440 – 391449
CAP Programator AB 391450 – 391459
Reichle+De-Massari AG 391460 – 391474
Swiss Bank Corp (London) 391475 – 391484
Unisys Enterprise Svr 391485 – 391489
Intel - Test Dev. Tech. 391490 – 391499
Ampex 391500 – 391755
391756 naas-spare
391757 naas-admin
391758 isps
391759 isps-admin
391760 mars
391761 mars-admin
391762 attcis_spare0
391763 attcis_spare1
391764 mail-server
391765 mail-server-spare
391766 attcis_spare2
391767 attcis_spare3
391768 attcis_spare4
391769 attcis_spare5
391770 attcis_spare6
391771 attcis_spare7
Integrated Systems, Inc. 391772 – 391779
Parametric Tech., Inc. 391780 – 391789
Ericsson Telecom AB 391790 – 391799
SLAC 391800 – 391849
391850 qhrdata
391851 qhrbackup
391852 minutedata
391853 prefecture
391854 supc
391855 suadmincrw
391856 suadminnotas
391857 sumessage
391858 sublock
391859 sumotd
staffware dev. (uk) 391860 - 391869
Staffware Dev. (UK) 391870 - 391879
391880 namesrvr
391881 disksrvr
391882 tapesrvr
391883 migsrvr
391884 pdmsrvr
391885 pvrsvr
391886 repacksrvr
391887 [unknown]

Convex Computer Corp. 391888 - 391951
391952 lookoutsrv
391953 lookoutagnt
391954 lookoutprxy
391955 lookoutsnmp
391956 lookoutrmon
391957 lookoutfut1
391958 lookoutfut2

windward 391959 - 391967
391968 sra_legato
391969 sra_legato_imgsrsrv
391970 sra_legato_0
391971 sra_legato_1
391972 sra_legato_2
391973 sra_legato_3
391974 sra_legato_4
391975 sra_legato_5
391976 sra_legato_6
391977 sra_legato_7
391978 sra_legato_8
391979 sra_legato_9

Brooktree Corp. 391980 - 391989
Cadence Design Systems 391990 - 391999
J. Frank & Associates 392000 - 392999
Cooperative Solutions 393000 - 393999
Xerox Corp. 394000 - 395023
395024 odbc_sqlretriever
3M 395025 - 395091
digital Zone Intl. 395092 - 395099
Software Professionals 395100 - 395159
Del Mar Solutions 395160 - 395164
395165 ife-es
395166 ife-resmgr
395167 ife-aes
395168 ife-bite
395169 ife-loader
395170 ife-satcom
395171 ife-seat
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>395172</td>
<td>ife-dbmgr</td>
<td></td>
</tr>
<tr>
<td>395173</td>
<td>ife-testmgr</td>
<td></td>
</tr>
<tr>
<td>395174</td>
<td>atrium_server</td>
<td></td>
</tr>
<tr>
<td>395175</td>
<td>ase_director</td>
<td></td>
</tr>
<tr>
<td>395176</td>
<td>ase_agent</td>
<td></td>
</tr>
<tr>
<td>395177</td>
<td>ase_hsm</td>
<td></td>
</tr>
<tr>
<td>395178</td>
<td>ase_mgr</td>
<td></td>
</tr>
<tr>
<td>395179</td>
<td>ase_sim</td>
<td></td>
</tr>
</tbody>
</table>

Hewlett-Packard 395180 - 395194
XES, Inc. 395195 - 395199
Unitech Products 395200 - 395249
TransSys 395250 - 395505
Unisys Govt Systems 395506 - 395519
Bellcore 395520 - 395529
IBM 395530 - 395561
AT&T Network Services 395562 - 395571
Data General 395572 - 395577
Swiss Bank Corp 395578 - 395597
Swiss Bank Corp 395598 - 395637
Novell 395638 - 395643
Computer Associates 395644 - 395650
Omneon Video Networks 395651 - 395656
unassigned 395657 - 395908
UK Post Office 395909 - 395924
AEROSPATIALE 395925 - 395944
Result d.o.o. 395945 - 395964
DataTools, Inc. 395965 - 395980
CADIS, Inc. 395981 - 395990
Cummings Group, Inc. 395991 - 395994
Cadre Technologies 395995 - 395999
American Airlines 396000 - 396999
Ericsson Telecom TM Div 397000 - 398023
IBM 398024 - 398028
Toshiba OME Works 398029 - 398033
TUSC Computer Systems 398034 - 398289
AT&T 398290 - 398320
Ontario Hydro 398321 - 398346
Micron Corporation 398347 - 398364
unassigned 398365 - 398591
Pegasystems, Inc. 398592 - 399616
Spectra Securities Soft 399617 - 399850
QualCom 399851 - 399866
unassigned 399867 - 399884
Altris Software Ltd. 399885 - 399899
ISO/IEC WG11 399900 - 399919
Parametric Technology 399920 - 399949
Dolby Laboratories 399950 - 399981
unassigned 399982 - 399991

Thurlow Standards Track [Page 60]
Xerox PARC                          399992 - 399999
#
Next Inc.                             200100000 - 200199999
Netwise (RPCtool)                      200200000
 Concurrent Computer Corp             200200001 - 200200007
AIM Technology                        200300000 - 200399999
TGV                                    200400000 - 200499999
#
# Sun-assigned authentication flavor numbers
#
AUTH_NONE                0       /* no authentication, see RFC 1831 */
                /* a.k.a. AUTH_NULL */
AUTH_SYS                 1       /* unix style (uid+gids), RFC 1831 */
                /* a.k.a. AUTH_UNIX */
AUTH_SHORT                2       /* short hand unix style, RFC 1831 */
AUTH_DH                  3       /* des style (encrypted timestamp) */
                /* a.k.a. AUTH_DES, see RFC 2695 */
AUTH_KERB                4       /* kerberos auth, see RFC 2695 */
AUTH_RSA                 5       /* RSA authentication */
RPCSEC_GSS                6       /* GSS-based RPC security for auth,
                integrity and privacy, RPC 5403 */
AUTH_NW                  30001   NETWARE
AUTH_SEC                 200000  TSIG NFS subcommittee
AUTH_ESV                  200004  SVr4 ES
AUTH_NQNFS               300000  Univ. of Guelph - Not Quite NFS
AUTH_GSSAPI              300001  OpenVision <john.linn@ov.com>
AUTH_ILU_UGEN            300002  Xerox <janssen@parc.xerox.com>
                - ILU Unsecured Generic Identity
#
# Small blocks are assigned out of the 39xxxx series of numbers
#
AUTH_SPNEGO               390000
390000 - 390255 NFS 'pseudo' flavors for RPCSEC_GSS
390003 - kerberos_v5 authentication, RFC 2623
390004 - kerberos_v5 with data integrity, RFC 2623
390005 - kerberos_v5 with data privacy, RFC 2623
2000000000  Reserved
2001000000  NeXT Inc.
Normative References


Informative References


Author’s Address

Robert Thurlow
Sun Microsystems, Inc.
500 Eldorado Boulevard, UBRM05-171
Broomfield, CO 80021

Phone: 877-718-3419
EMail: robert.thurlow@sun.com