

**6 March 1998**

**The Systems Management Abstract Service Type  
draft-ietf-srvloc-sysman-02.txt**

Status of This Memo

This document is a submission by the Service Location Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the [srvloc@corp.home.net](mailto:srvloc@corp.home.net) mailing list.

Distribution of this memo is unlimited.

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

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

To learn the current status of any Internet-Draft, please check the ``[1id-abstracts.txt](#)'' listing contained in the Internet-Drafts Shadow Directories on [ftp.is.co.za](ftp://ftp.is.co.za) (Africa), [nic.nordu.net](ftp://nic.nordu.net) (North Europe), [ftp.nis.garr.it](ftp://ftp.nis.garr.it) (South Europe), [munniari.oz.au](ftp://munniari.oz.au) (Pacific Rim), [ds.internic.net](ftp://ds.internic.net) (US East Coast), or [ftp.isi.edu](ftp://ftp.isi.edu) (US West Coast).

Abstract

This document presents definitions for the "sysman" (systems management) abstract service, and the "dmi" (desktop management interface), "cim" (common information model), "snmp" (simple network management protocol), and "host" concrete service types. These definitions are suitable for use with the Service Location Protocol [[1](#)].

Systems management, as used in this document, relates to the monitoring, diagnosing, service and support of desktop and host computers.

**[1](#). Introduction**

Service templates and abstract service types are defined in [[2](#)]. The systems management abstract service type is designed to organize information from various services, protocols, and schemas that relate to systems management.

Day

Expires 6 September 1998

[Page 1]

## **2. Using SLP For Network and Systems Management**

When deploying SLP for purposes of network or systems management, there are some important points to consider. These include ease of administration, having a low impact on the network, and providing SLP support for the entire range of systems on the network. (Note that using SLP to discover managed systems will only find managed systems that support SLP.)

Network and systems management applications need to continue working when other network components are broken. Further, management applications frequently need to obtain information that has a very short lifetime. These considerations motivate some of the recommendations that are listed below.

The Service Location Protocol can provide a number of applications for network and systems management. For example:

- A manager can use SLP to discover managed systems without performing brute-force scanning of the network.
- A manager can select manageable systems based on specific attributes, such as support for a specific management protocol, agent, instrumentation group, and software distribution format.
- A managed system can use SLP to find its manager(s), to which the managed system can then send traps and issue trouble tickets.
- A managed system can use SLP to publish basic information about itself, such as its operating system vendor, type and version; its MAC address(es); operator information; management agent(s) hosted by that system, and so on.
- A manager can periodically search for new hosts using SLP.

A good strategy to follow when using SLP for manageability includes the following points:

- Peer-to-peer implementation. Each managed system, along with each management application, implements UA and SA functionality. The system advertises its management services using the "service:sysman" concrete types.
- Bootstrap capability. Each managed system, along with each management application can begin functioning with no knowledge of existing manageability services beyond a knowledge of its own UA. For example, a management application can, with no foreknowledge, use SLP to build a list of manageable stations and their basic capabilities.

Day

Expires 6 September 1998

[Page 2]

This bootstrap capability requires the use of multicast or broadcast with convergence. Multicasting and broadcasting should be avoided when possible. However, bootstrap capability is a key aspect of systems management. (What happens when the DA is broken and needs to be managed?) Therefore, managers and managed stations should use DAs for discovery when they exist, but must also have the ability to perform multicast or broadcast with convergence.

- DAs for support of mobile and occasionally connected stations, as well as for scalability. DAs provide a good way to support the discovery of mobile and occasionally connected systems because they can act as SLP proxies for such systems, which may be connected via a slow link. In addition, DAs provide scalability to SLP.
- If the site implements administrative domains, it should also implement a management-specific domain. SAs that publish "service:sysman" types should register their information with DAs that are scoped for the management-specific domain.

More information on domain scoping for management is in [section 9](#) below, "Domain Scoping for Manageability."

### 3. The "sysman" Abstract Service

```
-----template begins here-----
type = service:sysman

version = 0.0

language = en

description =
    The 'service:sysman' abstract service type organizes information
    from various services, protocols, and schemas that relate to
    systems management. Systems management, as used in this
    document, relates to the monitoring, diagnosing, service and
    support of desktop and host computers.

url-syntax =
url-path   = hosturl / dmiurl / cimurl / snmpurl
hosturl    = url as defined in "Host Service Type" (below)
dmiurl     = url as defined in "DMI Service Type" (below)
cimurl     = url as defined in "CIM Service Type" (below)
snmp1url   = url as defined in "SNMPv1 Service Type" (below)
-----template ends here-----
```

contact="Erik Guttman" <Erik.Guttman@sun.com>

Day

Expires 6 September 1998

[Page 3]

security considerations=

Note the security considerations associated with the concrete types defined below.

#### 4. The Host Service Type

The "host" concrete service type provides information immediately relevant to the management of a desktop or host computer, such as computer's the operating system version and vendor, its physical location, and so on.

The service:sysman:host service template is as follows:

-----template begins here-----  
 type = service:sysman:host

version = 0.1

language = en

description =

The "host" concrete service type provides information immediately relevant to the management of a desktop or host computer.

url syntax=

```

url-path      = ([machine-name] os-name os-vendor
                  os-majorversion [os-minorversion]
                  [os-revision] mac-address [contact])
machine-name  = ";machine-name =" 1*alpha-num
os-name       = ";os-name =" 1*alpha-num
os-vendor     = ";os-vendor =" 1*alpha-num
os-majorversion = ";os-major =" 1*DIGIT
os-minorversion = ";os-minor =" 1*alpha-num
os-revision   = ";os-revision =" 1*alpha-num
mac-address   = ";mac-address =" 12HEXDIG
owner        = ";owner =" 1*CHAR
machine-uuid  = ";uuid =" as defined in [3]
                  ;uuid is a 128-bit value, represented
                  ;as a string, that provides a unique
                  ;id to hardware that supports this
                  ;this feature

```

; the fields beginning with machine-name and ending with  
 ; owner are each defined as an attributed, below.

machine-name = string 0 L

# The machine-name attribute is optional. It SHOULD be included  
 # whenever the computer has a network name other than a domain  
 # name.

Day

Expires 6 September 1998

[Page 4]



os-name = string L  
# The os-name attribute is required. It signifies the operating  
# system running on the computer. Although there is no list of  
# allowed values, the following values are recommended for their  
# operating systems:  
#  
# (solaris, netware, windows-nt, windows-95, windows,  
# linux, dos, macos, aix, irix, hpux, bsd, freebsd)  
#  
# Implementations SHOULD use a value from the preceeding list  
# when representing one of those operating systems. Other values  
# may be used for other operating systems.

os-vendor = string L  
# The os-vendor attribute is required. It signifies the vendor of  
# operating system running on the computer. For example, "Sun  
# Microsystems," "Microsoft," or "Santa Cruz Operation."  
#  
# NOTE: There is an important convention for representing  
# vendor strings. See [section 9](#).

os-majorversion = integer L  
# The os-majorversion attribute is required. It signifies the  
# major version of the operating system running on the computer.

os-minorversion = string 0 L  
# The os-minorversion attribute is optional. It signifies the minor  
# version of the operating system running on the computer.

os-revision = string 0 L  
# The os-revision attribute is optional. Some vendors use a  
# revision string to indicate the build of their operating systems,  
# instead of a version number.

mac-address = string M L  
# The mac-address attribute is required. It is a string  
# representation of the hexadecimal mac address of each network  
# card in the computer. There MUST be one mac-address attribute for  
# each network card in the computer.

owner = string 0  
# The owner attribute is optional and presents a human-readable  
# string containing contact information for the computer. This  
# attribute SHOULD contain a name and phone number, but MAY contain  
# other information.

machine-uuid = string 0 L  
# The machine-uuid attribute is optional. It signifies the machine  
# UUID or GUID of the host. This value is specific to the machine

```
# itself and not the host operating system nor to any of its  
# applications. For machines that do not have an embedded UUID, a  
# software program may generate a machine UUID.
```

Day

Expires 6 September 1998

[Page 5]

-----template ends here-----

contact="Michael Day" <michael\_day@ccm.ut.intel.com>

security considerations=

Free access to computer owner information may present a security risk in some environments. The other information provided by this service type is generally available in most environments.

## 5. The DMI Service Type

The "dmi" service type provides information about the computer's support of the Desktop Management Interface (DMI), an instrumentation standard supported by the Desktop Management Task Force (DMTF) [4].

The service:sysman:dmi service template is as follows:

-----template begins here-----

type = service:sysman:dmi

version = 0.1

language = en

description =

The "dmi" concrete service type provides information the availability of Desktop Management Interface services on the computer. For information on the desktop management interface, see <http://www.dmtf.org>

url syntax=

```
url-path      = (provider vendor provider-major
                  provider-minor [remote-interface]
                  [access-point] [security-level])
provider-vendor = ";vendor = " 1*alpha-num
provider-major  = ";major = " 1*DIGIT
provider-minor  = ";minor = " 1*alpha-num
remote-interface = ";remote=" 1*ALPHA"
access-point    = ";access-point = " 1*safe-char
security-level  = ";security = " 1*safe-char
; The fields beginning with provider-major and ending with
; security-level are defined as attributes, below
```

provider-vendor = string L

# The provider-vendor attribute is required. It signifies the  
# vendor of the DMI service provider running on the computer.  
#

# NOTE: There is an important convention for representing  
# vendor strings. See [section 9](#).

Day

Expires 6 September 1998

[Page 6]

```
provider-major = integer X
# The provider-major attribute is required, and SHOULD be included
# in service request predicates. It signifies the major version of
# the computer's DMI service provider. Allowable values are as
# follows:
```

```
1, 2
```

```
provider-minor = string L
# The provider-minor attribute is required. It signifies the minor
# revision of the computer's DMI service provider.
```

```
remote-interface = string L 0
# The remote-interface attribute is optional. It signifies the
# network interface to the DMI service provider. Allowable values
# are as follows:
```

```
DCE, ONC, TI, RAP
```

```
access-point = string L 0
ncacn_ip_tcp
# The access-point attribute is optional. It signifies the binding
# access point of the service provider's remote interface.
# Allowable values are as follows:
```

```
ncacn_ip_tcp      ; connection-oriented, TCP over IP
ncadg_ip_udp,     ; datagram, UDP over IP
ncacn_nb_tcp,     ; connection-oriented, NetBIOS over TCP
ncacn_spx,        ; connection-oriented, Novell SPX
ncadg_ipx         ; datagram, Novell IPX
```

```
security-level = string L 0
Off
# The security-level attribute is optional. It signifies the
# security mechanisms that are active with the computer's service
# provider.
On, Off
```

```
-----template ends here-----
```

```
contact="Michael Day" <michael_day@ccm.ut.intel.com>
```

```
security considerations=
```

```
  The Desktop Management Interface does not yet have a standard
  security scheme. Some attributes may not be secured. Information
  about remote access points to the DMI service provider may
  present a security risk if the RPC mechanism is not administered
  properly.
```

## **6. The CIM Service Type**

The "cim" service type provides information about the computer's

support of the Common Information Model (CIM)[\[5\]](#). CIM is a schema description language, meta-schema, and core schema published by the Desktop Management Task Force.

Day

Expires 6 September 1998

[Page 7]

The service:sysman:cim service template is as follows:

-----template begins here-----

type = service:sysman:cim

version = 0.0

language = en

description =

The "cim" concrete service type provides information the  
availability of Common Information Model services on the computer.  
For more information see <http://www.dmtf.org>

url syntax=

url-path = (cim-major cim-minor cim-revision om-vendor  
om-major om-minor om-revision  
[remote-interface] [access-point])

cim-major = ";cim-major =" 1\*DIGIT

cim-minor = ";cim-minor =" 1\*DIGIT

cim-revision = ";cim-revision =" 1\*alpha-num

om-vendor = ";om-vendor =" 1\*alpha-num

om-major = ";om-major =" 1\*DIGIT

om-minor = ";om-minor =" 1\*DIGIT

om-revision = ";om-revision =" 1\*alpha-num

remote-interface= ";remote =" 1\*alpha-num

access-point = ";access-point =" 1\*safe-char

; The fields beginning with cim-major and ending with  
; access-point defined as attributes, below.

cim-major = integer X

# The cim-major attribute is required and signifies the major  
# version of the Common Information Model Schema Description  
# Language that is supported on the computer. This attribute SHOULD  
# be included in service request predicates. Allowable values are  
# as follows:

1, 2

cim-minor = integer X

# The cim-minor attribute is required and signifies the minor  
# version of the Common Information Model Schema Description  
# Language that is supported on this computer. This attribute SHOULD  
# be included in service request predicates.

cim-revision = string L X

# The cim-revision attribute is required and signifies the  
# revision of the Common Information Model Schema Description  
# Language that is supported on this computer. This attribute  
# SHOULD be included in service request predicates.

Day

Expires 6 September 1998

[Page 8]



```
om-vendor = string L X
# The om-vendor attribute is required and signifies the vendor of
# the CIM object manager (CIMOM) that is present on the computer.
# This attribute SHOULD be included in service request predicates.
#
# NOTE: There is an important convention for representing
# vendor strings. See section 9.

om-major = integer X
# The om-major attribute is required and signifies the major
# version of the CIM object manager (CIMOM) that is supported on the
# computer. This attribute SHOULD be included in service request
# predicates.

om-minor = integer X
# The om-minor attribute is required and signifies the minor
# version of the CIM object manager (CIMOM) that is supported on the
# computer. This attribute SHOULD be included in service request
# predicates.

om-revision = string L
# The om-minor attribute is required and signifies the revision
# version of the CIM object manager (CIMOM) that is supported on the
# computer.

remote-interface = string L 0
# The remote-interface attribute is optional and signifies the
# network interface to the CIM object manager (CIMOM). Examples
# of expected values include be "DCE-RPC, "IIOP," or "DCOM."

access-point = string L 0
# The access-point attribute is optional and signifies the network
# service access point to the CIM object manager (CIMOM). Examples
# of expected values include "TCP/IP," and "IPX."

-----template ends here-----

contact="Michael Day" <michael_day@ccm.ut.intel.com>

security considerations=
  CIM security is derived from the CIMOM and the attributes defined
  in the CIM schema being accessed. Administrators should take care to
  understand the security attributes of the CIM schemas and the
  characteristics o their CIMOM before publishing information about
  remote access to CIM data.
```

## **7. The SNMPv1 Service Type**

The "snmpv1" service type provides information about Simple Network

Management Protocol version 1 services that are supported by the computer.

Day

Expires 6 September 1998

[Page 9]

The service:sysman:snmpv1 service template is as follows:

```
-----template begins here-----
type=service:sysman:snmpv1
version=0.0
lang=en

description=
  The 'service:sysman:SNMPv1:' URL provides information about the
  SNMPv1 manageability of a given host.  Namely, if this URL exists
  for a host (denoted by the <addr-spec> in the URL,) the host
  supports SNMPv1.

url-syntax=
  url-path      = ( [port-list] [comm-string] [oid-list])
                  ; None of the attributes listed in the URL path
                  ; are required.  They MAY be included.
  port-list     = ";ports=" port-list
  ports         = port / port "," ports
                  ; See the Service URL <port> production rule.
                  ; This field is defined as an attribute, below.
  comm-string   = ";read-community-string=" 1*uchar
                  ; See the Service URL <uchar> production rule.
                  ; This field is defined as an attribute, below.

ports=integer M L O
161
# This attribute must be included if the service:URL does not
# contain transport protocol information. It lists all UDP ports on
# which SNMP Agents are listening.

read-community-string=string L O
# The read community string may be included as an attribute of
# a service:network-managment:snmp: URL.  This is useful in
# cases where the community string is PUBLIC and ease of access
# to the SNMP Agent is desired.  See the 'security considerations.'

-----template ends here-----

contact="Erik Guttman" <Erik.Guttman@sun.com>

security considerations=
  The read-community-string MUST only be included if the value
  of this string is considered to be public.  If this attribute
  is included, absolutely anyone may access the SNMP Agent and
  get information from it.  This may be desirable in some cases.
  If this string is considered confidential information, the
  read-community-string MUST NOT be included in the URL path
  nor in service registrations of the URL made through SLP or
```

other protocols.

Day

Expires 6 September 1998

[Page 10]

## **8. Other Security Considerations**

In addition to specific security considerations list above, each of these service types inherit the security considerations of the "service:" URL scheme as specified in [2].

## **9. Domain Scoping for Manageability**

When a site activates SLP administrative scoping, this can cause problems for management applications that use SLP for discovery.

The problems are associated with visibility of published services. Specifically, if a management application is not part of an administrative scope, it will not be able to discover management services that are within that scope.

Management applications sometimes have a peculiar need to "see" and "hear" everything that exists in a computing environment, and this need extends to service location.

Therefore, whenever SLP administrative scoping is in place, there should also be a service-specific scope established for the "service:sysman" type, for example "systems-management".

With a systems-management scope in place, all UAs and SAs that support the "service:sysman" type will need to be configured to use that scope to advertise and discover management services. SLP supports scope lists, meaning that UAs and SAs can be members of more than one scope. In addition, the SLP DHCP discovery options will support an optional scope list.

This will allow for management applications to have global vision even when administrative scoping divides the computing environment into disjoint scopes.

## **10. Core Rules**

**This document uses the following core rules in its grammar definitions:**

alpha-num	= ALPHA / DIGIT / "-" / "."
safe-char	= as defined in [2]
DIGIT	= as defined in [7]
ALPHA	= as defined in [7]
HEXDIG	= as defined in [7]
CHAR	= as defined in [7]

## **11. Vendor Strings**

In order to allow programmatic searching on the vendor attribute,  
there must be a convention for representing vendor strings.

Day

Expires 6 September 1998

[Page 11]

To illustrate the problem, imagine a vendor attribute is registered as follows:

```
"vendor=Acme Widgets, Limited"
```

Subsequently, three separate service requests are issued:

- (1) SrvRqst: "vendor=Acme"
- (2) SrvRqst: "vendor=Acme Ltd."
- (3) SrvRqst: "vendor=Acme Widgets, Limited"

Only the third request above will succeed.

One solution to this problem is to have a list of allowable values for vendor names. However, this is unrealistic because it limits vendors to a predefined small set.

The better solution is to use a simple convention for representing vendor names:

Vendor names SHOULD be represented by using the proper or trademarked name of the vendor with no embellishments.

For example:

Sun Microsystems, Wind River Systems,  
International Business Machines, Amazon.com,  
Digital Equipment Corporation, Hewlett-Packard

The convention is to omit qualifiers such as "Inc.," "Ltd.," and so on, unless those qualifiers are part of the vendors trademarked or proper name. While this convention is not foolproof, it should allow programmatic matching of vendors in almost all cases.

## **12. Acknowledgments**

**This document benefited from the insights and work of Andrea Westerinen, John Keith, John Kilfoil. Erik Guttman was a contributing author of the systems management templates.**

Day

Expires 6 September 1998

[Page 12]



### **13. References**

Request For Comments (RFC) and Internet Drafts documents are available from <URL:ftp://ftp.internic.net> and numerous mirror sites

- [1] E. Guttman, C. Perkins, J. Veizades, M. Day, "Service Location Protocol version 2.02," Internet Draft (work in progress), January 1998.
- [2] C. Perkins, E. Guttman, J. Kempf, "Service Templates and 'service:' Schemes, version 7" Internet Draft (work in progress), February 1998.
- [3] P. Leach, R. Salz, "UUIDs and GUIDs" Internet Draft (work in progress), February 1998.
- [4] Desktop Management Interface Specification 2.0 - <http://www.dmtf.org/>
- [5] Desktop Management Task Force, "Common Information Model Specification," version 2.0e, December 1997. <http://www.dmtf.org>
- [6] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Locators (URL): Generic Syntax and Semantics," [RFC1738](#) as amended by [RFC1808](#) and updated by [draft-fielding-url-syntax-09.txt](#), May 1997. (work in progress).
- [7] D. Crocker and P. Overell. Augmented BNF for Syntax Specifications: ABNF. [RFC 2234](#), November 1997.

### **14. Author's address**

Michael Day  
Intel  
734 East Utah Valley Drive, ste. 300  
American Fork, UT 84003  
USA

Phone: +1 801 763-2341  
Fax: +1 801 756-8349  
EMail: [michael.day@intel.com](mailto:michael.day@intel.com)