

## Definitions of Managed Objects for WWW Services

### 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) The Internet Society (1999). All Rights Reserved.

### Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet Community. In particular it describes a set of objects for managing World Wide Web (WWW) services.

### Table of Contents

1 Introduction .....	1
2 The SNMP Management Framework .....	2
3 Terminology .....	3
4 Overview .....	4
4.1 Purpose and Requirements .....	4
4.2 Relationship to other Standards Efforts .....	5
4.3 WWW Services .....	5
4.4 Document Transfer Protocol .....	6
5 Structure of the MIB .....	7
5.1 Service Information Group .....	7
5.2 Protocol Statistics Group .....	7
5.3 Document Statistics Group .....	8
6 Definitions .....	10
7 Document Transfer Protocol Mappings .....	36
7.1 The HyperText Transfer Protocol .....	36
7.2 The File Transfer Protocol .....	37
8 Security Considerations .....	38
9 Intellectual Property .....	39
10 Acknowledgments .....	39

11 Editors' Addresses .....	39
12 References .....	40
13 Full Copyright Statement .....	43

## 1. Introduction

This memo defines a set of objects for managing World Wide Web (WWW) services. This MIB extends the application management framework defined by the System Application Management MIB (SYSAPPL-MIB) [23] and the Application Management MIB (APPLICATION-MIB) [24]. The MIB is also self-contained so that it can be implemented and used without having to implement or install the APPLICATION-MIB or the SYSAPPL-MIB.

The protocol statistics defined in the WWW Service MIB are based on an abstract document transfer protocol (DTP). This memo also defines a mapping of the abstract DTP to HTTP and FTP. Additional mappings may be defined in the future in order to use this MIB with other document transfer protocols. It is anticipated that such future mappings will be defined in separate RFCs.

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 RFC 2119 [17].

## 2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2571 [1].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIV2, is described in STD 58, RFC 2578 [5], RFC 2579 [6] and RFC 2580 [7].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].

- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- o A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

### 3. Terminology

This section defines the terminology used throughout this document.

- o The 'World Wide Web' (WWW) is a world wide information system which is based on the concept of documents that are linked together by embedding references (links) to other local or remote documents.
- o A 'document' is a coherent piece of data which is accessible in the World Wide Web. No assumptions are made about the content or the type of a document.
- o A 'Uniform Resource Locator' (URL) is a formatted string representation for a document available via the Internet. URLs are used to express references between documents. For the syntax and semantics of the URL string representation refer to RFC 2396 [18]
- o A 'Document Transfer Protocol' (DTP) is a protocol used within the World Wide Web to invoke actions on documents. The DTP is an abstraction from real protocols, such as HTTP [19,20] or FTP [21].

- o A 'request' is a DTP protocol operation which is targeted to a 'document' and invokes an action on the target document. The request type specifies the action that should be performed. A request can have a document associated with it.
- o A 'response' is a DTP protocol operation which is returned as a result of a previous (and associated) request. The response status indicates if the requested action was successful or if errors occurred. A response can have a document associated with it.
- o A 'WWW service' is a set of actions that can be invoked on a document. Typical actions are the transfer of documents or the retrieval of administrative information about documents. WWW services are provided by means of a DTP. A WWW service can be identified by the DTP protocol used to invoke services and the transport endpoint used by that protocol.
- o A 'client' is a program which establishes connections for the purpose of sending requests and receiving responses.
- o A 'server' is a program that accepts connections in order to service requests by sending back responses.
- o A 'proxy' is an intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them on, with possible translation, to other servers.
- o A 'caching proxy' is a proxy with the capability of locally storing responses to associated requests. A caching proxy can respond to similar requests with a previously stored response.

#### 4. Overview

The World Wide Web (WWW) is a global network of information. Information is stored in documents, which can have various formats, including hyper-text and multi-media documents. Access to these documents is provided by servers which are located all around the world and are linked to each other via hyper-links embedded in documents.

The usability of the World Wide Web depends largely on the performance of the services realized by these servers. The services are typically monitored through log files. This becomes a difficult task when a single organization is responsible for a large number of services. It is therefore desirable to treat WWW services as objects that can be managed by using the Internet network management framework [22].

#### 4.1. Purpose and Requirements

The goal of this MIB is to define a standardized set of objects which lead to integrated and improved performance and fault management in a heterogeneous environment of WWW services. This MIB focuses on the service-oriented view. It does not deal with the process oriented view, which is covered by the System Application MIB [23] and the Application MIB [24].

This document defines a set of managed objects to monitor WWW services for short-term operational purposes, such as problem detection and troubleshooting. No attempts are made here to cover accounting or hit metering issues.

The scope of the MIB is further limited by the requirement that an implementation conforming to this MIB must be possible without putting a huge CPU or memory burden on the WWW server implementation.

In addition, this MIB does not cover WWW service configuration. Server software has become an open market where competing vendors constantly invent new features in order to shape their products. It is therefore not possible to reach consensus on a common way to configure WWW services at this point in time.

#### 4.2. Relationship to other Standards Efforts

The WWW Service MIB fits into the application management architecture defined in the System Application MIB [23]. The System Application MIB and the Application MIB [24] use a process-oriented view, where an application is viewed as a collection of processes. The WWW Service MIB described in this memo uses a service-oriented view, which looks at the services provided by a set of processes.

The relationship between the process-oriented view and the service-oriented view is a many-to-many relationship, because one process can implement multiple services and multiple services can be implemented by a single set of processes. The Application Management MIB [24] contains generic mapping tables, which map back and forth between both views.

The WWW Service MIB interfaces to the Application MIB [24] by using the service instance identifier (applSrvIndex) for wwwServiceIndex if an applicable instance of applSrvIndex is available. The WWW Service MIB is self-contained and can be implemented as a stand-alone module if the service-level tables in the Application MIB are not available.

#### 4.3. WWW Services

The MIB is organized around the concept of WWW services. WWW services are a set of actions that can be invoked on a document. A WWW service is provided or used by either a client, a server or a proxy. Clients send out requests for information to server or proxy server. Servers receive, process and respond to requests received from clients. Servers usually have access to local documents, which can be transferred to clients.

A proxy is a special server, who acts as both a server and a client for the purpose of making requests on behalf of other clients. A proxy is able to translate between the client and the origin server. A proxy might also interact with other information retrieval system, like for example databases.

The MIB defined in this memo distinguishes between outgoing and incoming requests and responses. This makes it possible to obtain statistics for clients, servers and proxies with a single set of objects.

A special proxy server is the caching proxy, which maintains a cache of previously received documents in order to reduce the bandwidth used by World Wide Web clients. One interesting piece of management information is the percentage of requests that were served from the cache of the caching proxy (hits/miss-ratio). This ratio is not contained explicitly in this MIB. Instead, the ratio can be derived from the objects that count incoming and outgoing requests and responses.

#### 4.4. Document Transfer Protocol

The MIB is based on the concept of an abstract document transfer protocol (DTP). The purpose of the abstract document transfer protocol is to make the MIB definitions independent from concrete protocols, like the Hypertext Transfer Protocol (HTTP) [19,20] or the File Transfer Protocol (FTP) [21].

The abstract document transfer protocol makes the following assumptions about a concrete transfer protocol:

- o The transfer protocol uses a request/response style of interactions.
- o Every request contains a request type, which defines the operations performed by the receiving server. The request type is represented by an OCTET STRING. It might be necessary to define a translation into an OCTET STRING value for protocols that use numbers to identify request types.

- o A response contains a status code, which indicates if the request was processed successfully or which error occurred. The status code is represented as an INTEGER value. It might be necessary to define a mapping for protocols that do not use an INTEGER status code.
- o A transfer protocol can send multiple responses for a single request. Multiple responses are counted separately in the protocol statistics group.

A primary response has to be identified for the document statistics. The primary response is the response that indicates whether the request was successful.

Section 7 of this memo defines a mapping of the document transfer protocol to the HTTP protocol and the FTP protocol. Mappings to other protocols, like NNTP [25] or WebNFS [26,27] might be defined in the future.

## 5. Structure of the MIB

This section presents the structure of the MIB. The objects are arranged into the following groups:

- o service information
- o protocol statistics
- o document statistics

### 5.1. Service Information Group

The service information group consists of a single table describing all the WWW services managed by the SNMP agent. The service table contains administrative network management information for (potentially) multiple WWW services running on a single host. It also contains information for all services within virtual domains of a host. The columnar objects in the table can be divided into two main groups:

- o global administrative information of the service, such as service contact person, and
- o network information, such as the transfer protocol.

## 5.2. Protocol Statistics Group

The protocol statistics group provides network management information about the traffic received or transmitted by a WWW service. This group contains counters related to DTP protocol operations and consists of five tables:

- o The `wwwSummaryTable` contains a set of network traffic related counters. The table provides a summarization of the network traffic and protocol operations related to a WWW service. It is well recognized that certain variables are redundant with respect to the request and response tables, but they are added to provide an operator a quick overview and to reduce SNMP network traffic.
- o The `wwwRequestInTable` contains detailed information about incoming requests. Every particular request type is counted separately.
- o The `wwwRequestOutTable` contains detailed information about outgoing requests. Every particular request type is counted separately.
- o The `wwwResponseInTable` contains detailed information about incoming responses. Every particular response type is counted separately.
- o The `wwwResponseOutTable` contains detailed information about outgoing responses. Every particular response type is counted separately.

## 5.3. Document Statistics Group

The document group contains information about the documents which were accessed in the past. The group provides four types of statistics.

1. Details about the last N attempts to invoke actions on documents.
2. The Top N documents sorted by the number of actions invoked on them computed over a time interval.
3. The Top N documents sorted by the number of content bytes transferred computed over a time interval.
4. Summary statistics computed over a time interval.



The Top N document statistics are collected in buckets in order to reduce agent resources and to allow a manager to detect changes in the service usage pattern. Buckets are filled over a configurable time interval. The agent computes the Top N statistics and starts a new bucket once the time interval for the bucket has passed. The time interval is configurable for each WWW service.

The document statistics group associates a response type to the request which invoked an action. In case a DTP sends multiple responses, the primary response must be used to derive the response type of the request/response interaction.

The group consist of the following tables:

- o The `wwwDocCtrlTable` provides the manager a means to limit the document statistic tables in size and to control the expiration and creation of buckets.
- o The `wwwDocLastNTable` provides the manager information about the last N documents which where accessed. The table lists the documents for which access was attempted along with the request and response type of the DTP and a status message. The request and response types provide a manager information of how attempts to invoke actions were handled by the DTP. The status message object provides human readable text to further describe the response type.

The number of documents in the `wwwDocLastNTable` is controlled by the `wwwDocCtrlLastNSize` object in the `wwwDocCtrlTable`. The `wwwDocCtrlLastNLock` object of the `wwwDocCtrlTable` allows a management application to lock the `wwwDocLastNTable` in order to retrieve a consistent snapshot of the fast changing `wwwDocLastNTable`.

- o The `wwwDocBucketTable` lists the buckets of statistical information that have been collected. An entry in the `wwwDocBucketTable` contains the creation timestamp of the bucket as well as summary information (number of accesses, number of documents accessed and number of bytes transferred).

The time interval is controlled by the `wwwDocCtrlBucketTimeInterval` object of the `wwwDocCtrlTable`. The maximum number of buckets maintained by the SNMP agent for a particular WWW service is controlled by the `wwwDocCtrlBuckets` object of the `wwwDocCtrlTable`.

- o The `wwwDocAccessTopNTable` provides the manager an overview of the top N documents which were accessed while statistics were collected for a particular bucket. The `wwwDocAccessTopNTable` is

sorted by the number of read attempts per document. The maximum number of entries in the `wwwDocAccessTopNTable` is controlled by the `wwwDocCtrlTopNSize` object.

- o The `wwwDocBytesTopNTable` provides the manager an overview of the top N documents which caused most of the network traffic while statistics were collected for a particular bucket. The `wwwDocBytesTopNTable` is sorted by the number of bytes transferred. The maximum number of entries in the `wwwDocBytesTopNTable` is controlled by the `wwwDocCtrlTopNSize` object.

The Top N statistics and the parameters of the underlying bucket are not visible in the MIB as long as the bucket is filling up. Instead, the following steps must be taken when the time interval for a bucket has passed:

1. A new entry in the `wwwDocBucketTable` is created to summarize the document statistics for that time interval.
2. The corresponding entries in the `wwwDocAccessTopNTable` and the `wwwDocBytesTopNTable` are computed and made available.
3. If the resulting number of entries in the `wwwDocBucketTable` for the WWW service now exceeds `wwwDocCtrlBuckets`, then the oldest bucket for this WWW service and all corresponding entries in the `wwwDocBucketTable`, `wwwDocAccessTopNTable`, and `wwwDocBytesTopNTable` are deleted.

Note that a bucket usually contains much more data than displayed in the Top N tables. The number of entries in the Top N table for a bucket is controlled by `wwwDocCtrlTopNSize`, while the number of entries in a bucket depends on the number of actions invoked on documents within the time interval over which a bucket is filled up. It is therefore suggested to discard the data associated with a bucket once the entries for the `wwwDocBucketTable`, `wwwDocAccessTopNTable` and `wwwDocBytesTopNTable` have been calculated.

## 6. Definitions

```
WWW-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, mib-2,  
    Counter32, Counter64, Integer32, Unsigned32, TimeTicks  
    FROM SNMPv2-SMI
```

TEXTUAL-CONVENTION, DisplayString, DateAndTime, TimeInterval  
FROM SNMPv2-TC

MODULE-COMPLIANCE, OBJECT-GROUP  
FROM SNMPv2-CONF

Utf8String  
FROM SYSAPPL-MIB;

wwwMIB MODULE-IDENTITY

LAST-UPDATED "9902251400Z"

ORGANIZATION "IETF Application MIB Working Group"

CONTACT-INFO

" Harrie Hazewinkel

Postal: Joint Research Centre of the E.C.  
via Fermi - Ispra 21020 (VA)  
Italy

Tel: +39+(0)332 786322

Fax: +39+(0)332 785641

E-mail: harrie.hazewinkel@jrc.it

Carl W. Kalbfleisch

Postal: Verio, Inc.  
1950 Stemmons Freeway  
Suite 2006  
Dallas, TX 75207  
US

Tel: +1 214 290-8653

Fax: +1 214 744-0742

E-mail: cwk@verio.net

Juergen Schoenwaelder

Postal: TU Braunschweig  
Bueltenweg 74/75  
38106 Braunschweig  
Germany

Tel: +49 531 391-3683

Fax: +49 531 489-5936

E-mail: schoenw@ibr.cs.tu-bs.de"

DESCRIPTION

"This WWW service MIB module is applicable to services  
realized by a family of 'Document Transfer Protocols'  
(DTP). Examples of DTPs are HTTP and FTP."

```
-- revision history

REVISION      "9902251400Z"
DESCRIPTION   "Initial version, published as RFC2594."

 ::= { mib-2 65 }

--
-- Object Identifier Assignments
--

wwwMIBObjects      OBJECT IDENTIFIER ::= { wwwMIB 1 }
wwwMIBConformance OBJECT IDENTIFIER ::= { wwwMIB 2 }

--
-- Textual Conventions
--

WwwRequestType ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The WwwRequestType defines the textual identification of
        request types used by a document transfer protocol. For
        the proper values for a given DTP, refer to the protocol
        mappings for that DTP."
    SYNTAX       OCTET STRING (SIZE (1..40))

WwwResponseType ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The WwwResponseType defines the different response values
        used by document transfer protocols. For the proper values
        for a given DTP, refer to the protocol mappings for that
        DTP."
    SYNTAX       Integer32 (0..2147483647)

WwwOperStatus ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The operational status of a WWW service. 'down' indicates
        that the service is not available. 'running' indicates
        that the service is operational and available. 'halted'
        indicates that the service is operational but not
        available. 'congested' indicates that the service is
        operational but no additional inbound associations can be
        accommodated. 'restarting' indicates that the service is
        currently unavailable but is in the process of restarting
        and will be available soon."
    SYNTAX       INTEGER {
                    down(1),
```

```

        running(2),
        halted(3),
        congested(4),
        restarting(5)
    }

WwwDocName ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "255a"
    STATUS      current
    DESCRIPTION
        "The server relative name of a document. If the URL were
        http://www.x.org/standards/search/search.cgi?string=test
        then the value of this textual convention would resolve
        to '/standards/search/search.cgi'. This textual convention
        uses the character set for URIs as defined in RFC 2396
        section 2."
    SYNTAX      OCTET STRING (SIZE (0..255))

-- The WWW Service Information Group
--
-- The WWW service information group contains information about
-- the WWW services known by the SNMP agent.

wwwService OBJECT IDENTIFIER ::= { wwwMIBObjects 1 }

wwwServiceTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF WwwServiceEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "The table of the WWW services known by the SNMP agent."
    ::= { wwwService 1 }

wwwServiceEntry OBJECT-TYPE
    SYNTAX      WwwServiceEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "Details about a particular WWW service."
    INDEX       { wwwServiceIndex }
    ::= { wwwServiceTable 1 }

WwwServiceEntry ::= SEQUENCE {
    wwwServiceIndex      Unsigned32,
    wwwServiceDescription Utf8String,
    wwwServiceContact     Utf8String,
    wwwServiceProtocol    OBJECT IDENTIFIER,
    wwwServiceName        DisplayString,
    wwwServiceType        INTEGER,

```

```
    wwwServiceStartTime      DateAndTime,
    wwwServiceOperStatus     WwwOperStatus,
    wwwServiceLastChange     DateAndTime
}
```

wwwServiceIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An integer used to uniquely identify a WWW service. The value must be the same as the corresponding value of the applSrvIndex defined in the Application Management MIB (APPLICATION-MIB) if the applSrvIndex object is available. It might be necessary to manually configure sub-agents in order to meet this requirement."

::= { wwwServiceEntry 1 }

wwwServiceDescription OBJECT-TYPE

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Textual description of the WWW service. This shall include at least the vendor and version number of the application realizing the WWW service. In a minimal case, this might be the Product Token (see RFC 2068) for the application."

::= { wwwServiceEntry 2 }

wwwServiceContact OBJECT-TYPE

SYNTAX Utf8String

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The textual identification of the contact person for this service, together with information on how to contact this person. For instance, this might be a string containing an email address, e.g. '<webmaster@domain.name>'."

::= { wwwServiceEntry 3 }

wwwServiceProtocol OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An identification of the primary protocol in use by this service. For Internet applications, the IANA maintains a registry of the OIDs which correspond to well-known application protocols. If the application protocol is not listed in the registry, an OID value of the form

{applTCPProtoID port} or {applUDPProtoID port} are used for TCP-based and UDP-based protocols, respectively. In either case 'port' corresponds to the primary port number being used by the protocol."

## REFERENCE

"The OID values applTCPProtoID and applUDPProtoID are defined in the NETWORK-SERVICES-MIB (RFC 2248)."

::= { wwwServiceEntry 4 }

## wwwServiceName OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The fully qualified domain name by which this service is known. This object must contain the virtual host name if the service is realized for a virtual host."

::= { wwwServiceEntry 5 }

## wwwServiceType OBJECT-TYPE

SYNTAX INTEGER {  
     wwwOther(1),  
     wwwServer(2),  
     wwwClient(3),  
     wwwProxy(4),  
     wwwCachingProxy(5)  
 }

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The application type using or realizing this WWW service."

::= { wwwServiceEntry 6 }

## wwwServiceStartTime OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The date and time when this WWW service was last started. The value SHALL be '0000000000000000'H if the last start time of this WWW service is not known."

::= { wwwServiceEntry 7 }

## wwwServiceOperStatus OBJECT-TYPE

SYNTAX WwwOperStatus

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"Indicates the operational status of the WWW service."

::= { wwwServiceEntry 8 }

```

wwwServiceLastChange OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The date and time when this WWW service entered its current
        operational state. The value SHALL be '0000000000000000'H if
        the time of the last state change is not known."
    ::= { wwwServiceEntry 9 }

-- The WWW Protocol Statistics Group
--
-- The WWW protocol statistics group contains statistics about
-- the DTP requests and responses sent or received.

wwwProtocolStatistics OBJECT IDENTIFIER ::= { wwwMIBObjects 2 }

wwwSummaryTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF WwwSummaryEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The table providing overview statistics for the
        WWW services on this system."
    ::= { wwwProtocolStatistics 1 }

wwwSummaryEntry OBJECT-TYPE
    SYNTAX      WwwSummaryEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Overview statistics for an individual service."
    INDEX       { wwwServiceIndex }
    ::= { wwwSummaryTable 1 }

WwwSummaryEntry ::= SEQUENCE {
    wwwSummaryInRequests      Counter32,
    wwwSummaryOutRequests     Counter32,
    wwwSummaryInResponses     Counter32,
    wwwSummaryOutResponses    Counter32,
    wwwSummaryInBytes         Counter64,
    wwwSummaryInLowBytes      Counter32,
    wwwSummaryOutBytes        Counter64,
    wwwSummaryOutLowBytes     Counter32
}

wwwSummaryInRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only

```



```
STATUS      current
DESCRIPTION
    "The number of requests successfully received."
 ::= { wwwSummaryEntry 1 }

wwwSummaryOutRequests OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of requests generated."
 ::= { wwwSummaryEntry 2 }

wwwSummaryInResponses OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of responses successfully received."
 ::= { wwwSummaryEntry 3 }

wwwSummaryOutResponses OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of responses generated."
 ::= { wwwSummaryEntry 4 }

wwwSummaryInBytes OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of content bytes received."
 ::= { wwwSummaryEntry 5 }

wwwSummaryInLowBytes OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The lowest thirty-two bits of wwwSummaryInBytes."
 ::= { wwwSummaryEntry 6 }

wwwSummaryOutBytes OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```

        "The number of content bytes transmitted."
 ::= { wwwSummaryEntry 7 }

wwwSummaryOutLowBytes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The lowest thirty-two bits of wwwSummaryOutBytes."
 ::= { wwwSummaryEntry 8 }

-- The WWW request tables contain detailed information about
-- requests send or received by WWW services.

wwwRequestInTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF WwwRequestInEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table providing detailed statistics for requests
         received by WWW services on this system."
 ::= { wwwProtocolStatistics 2 }

wwwRequestInEntry OBJECT-TYPE
    SYNTAX      WwwRequestInEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Request statistics for an individual service."
    INDEX       { wwwServiceIndex, wwwRequestInIndex }
 ::= { wwwRequestInTable 1 }

WwwRequestInEntry ::= SEQUENCE {
    wwwRequestInIndex      WwwRequestType,
    wwwRequestInRequests   Counter32,
    wwwRequestInBytes      Counter32,
    wwwRequestInLastTime   DateAndTime
}

wwwRequestInIndex OBJECT-TYPE
    SYNTAX      WwwRequestType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The particular request type the statistics apply to."
 ::= { wwwRequestInEntry 1 }

wwwRequestInRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only

```

```

STATUS      current
DESCRIPTION
    "The number of requests of this type received by this
    WWW service."
 ::= { wwwRequestInEntry 2 }

wwwRequestInBytes OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of content bytes per request type received
    by this WWW service."
 ::= { wwwRequestInEntry 3 }

wwwRequestInLastTime OBJECT-TYPE
SYNTAX      DateAndTime
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The date and time when the last byte of the last complete
    request of this type was received by this WWW service. The
    value SHALL be '0000000000000000'H if no request of this
    type has been received yet."
 ::= { wwwRequestInEntry 4 }

wwwRequestOutTable OBJECT-TYPE
SYNTAX      SEQUENCE OF WwwRequestOutEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The table providing detailed statistics for requests
    generated by the services on this system."
 ::= { wwwProtocolStatistics 3 }

wwwRequestOutEntry OBJECT-TYPE
SYNTAX      WwwRequestOutEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Request statistics for an individual service."
INDEX       { wwwServiceIndex, wwwRequestOutIndex }
 ::= { wwwRequestOutTable 1 }

WwwRequestOutEntry ::= SEQUENCE {
    wwwRequestOutIndex      WwwRequestType,
    wwwRequestOutRequests   Counter32,
    wwwRequestOutBytes      Counter32,
    wwwRequestOutLastTime   DateAndTime
}

```

```
wwwRequestOutIndex OBJECT-TYPE
    SYNTAX      WwwRequestType
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The particular request type the statistics apply to."
    ::= { wwwRequestOutEntry 1 }

wwwRequestOutRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of requests of this type generated by this
         WWW service."
    ::= { wwwRequestOutEntry 2 }

wwwRequestOutBytes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of content bytes per requests type generated
         by this WWW service."
    ::= { wwwRequestOutEntry 3 }

wwwRequestOutLastTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The date and time when the first byte of the last request
         of this type was send by this WWW service. The value SHALL
         be '0000000000000000'H if no request of this type has been
         send yet."
    ::= { wwwRequestOutEntry 4 }

-- The WWW response tables contain detailed information about
-- responses sent or received by WWW services.

wwwResponseInTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF WwwResponseInEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The table providing detailed statistics for responses
         received by WWW services on this system."
    ::= { wwwProtocolStatistics 4 }

wwwResponseInEntry OBJECT-TYPE
```

```

SYNTAX      WwwResponseInEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Response statistics for an individual service."
INDEX       { wwwServiceIndex, wwwResponseInIndex }
 ::= { wwwResponseInTable 1 }

```

```

WwwResponseInEntry ::= SEQUENCE {
    wwwResponseInIndex      WwwResponseType,
    wwwResponseInResponses  Counter32,
    wwwResponseInBytes      Counter32,
    wwwResponseInLastTime   DateAndTime
}

```

```

wwwResponseInIndex OBJECT-TYPE
    SYNTAX      WwwResponseType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The particular response type the statistics apply to."
    ::= { wwwResponseInEntry 1 }

```

```

wwwResponseInResponses OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of responses of this type received by this
         WWW service."
    ::= { wwwResponseInEntry 2 }

```

```

wwwResponseInBytes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of content bytes per response type received
         by this WWW service."
    ::= { wwwResponseInEntry 3 }

```

```

wwwResponseInLastTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The date and time when the last byte of the last complete
         response of this type was received by this WWW service. The
         value SHALL be '0000000000000000'H if no response of this
         type has been received yet."

```

```

 ::= { wwwResponseInEntry 4 }

wwwResponseOutTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF WwwResponseOutEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table providing detailed statistics for responses
         generated by services on this system."
    ::= { wwwProtocolStatistics 5 }

wwwResponseOutEntry OBJECT-TYPE
    SYNTAX      WwwResponseOutEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Response statistics for an individual service."
    INDEX       { wwwServiceIndex, wwwResponseOutIndex }
    ::= { wwwResponseOutTable 1 }

WwwResponseOutEntry ::= SEQUENCE {
    wwwResponseOutIndex      WwwResponseType,
    wwwResponseOutResponses  Counter32,
    wwwResponseOutBytes      Counter32,
    wwwResponseOutLastTime   DateAndTime
}

wwwResponseOutIndex OBJECT-TYPE
    SYNTAX      WwwResponseType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The particular response type the statistics apply to."
    ::= { wwwResponseOutEntry 1 }

wwwResponseOutResponses OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of responses of this type generated by this
         WWW service."
    ::= { wwwResponseOutEntry 2 }

wwwResponseOutBytes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of content bytes per response type generated

```

```

        by this WWW service."
 ::= { wwwResponseOutEntry 3 }

wwwResponseOutLastTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The date and time when the first byte of the last response of
        this type was sent by this WWW service. The value SHALL be
        '0000000000000000'H if response of this type has been send
        yet."
 ::= { wwwResponseOutEntry 4 }

-- The WWW Document Statistics Group
--
-- The WWW document statistics group contains statistics about
-- document read attempts.

wwwDocumentStatistics OBJECT IDENTIFIER ::= { wwwMIBObjects 3 }

wwwDocCtrlTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF WwwDocCtrlEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A table which controls how the MIB implementation
        collects and maintains document statistics."
 ::= { wwwDocumentStatistics 1 }

wwwDocCtrlEntry OBJECT-TYPE
    SYNTAX      WwwDocCtrlEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry used to configure the wwwDocLastNTable,
        the wwwDocBucketTable, the wwwDocAccessTopNTable,
        and the wwwDocBytesTopNTable."
    INDEX       { wwwServiceIndex }
 ::= { wwwDocCtrlTable 1 }

WwwDocCtrlEntry ::= SEQUENCE {
    wwwDocCtrlLastNSize      Unsigned32,
    wwwDocCtrlLastNLock      TimeTicks,
    wwwDocCtrlBuckets        Unsigned32,
    wwwDocCtrlBucketTimeInterval TimeInterval,
    wwwDocCtrlTopNSize       Unsigned32
}

```

**wwwDocCtrlLastNSize OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The maximum number of entries in the wwwDocLastNTable."

DEFVAL { 25 }

::= { wwwDocCtrlEntry 1 }

**wwwDocCtrlLastNLock OBJECT-TYPE**

SYNTAX TimeTicks

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"This object allows a manager to lock the wwwDocLastNTable in order to retrieve the wwwDocLastNTable in a consistent state. The agent is expected to take a snapshot of the wwwDocLastNTable when it is locked and to continue updating the real wwwDocLastNTable table so that recent information is available as soon as the wwwDocLastNTable is unlocked again.

Setting this object to a value greater than 0 will lock the table. The timer ticks backwards until it reaches 0. The table unlocks automatically once the timer reaches 0 and the timer stops ticking.

A manager can increase the timer to request more time to read the table. However, any attempt to decrease the timer will fail with an inconsistentValue error. This rule ensures that multiple managers can simultaneously lock and retrieve the wwwDocLastNTable. Note that managers must cooperate in using wwwDocCtrlLastNLock. In particular, a manager MUST not keep the wwwDocLastNTable locked when it is not necessary to finish a retrieval operation."

::= { wwwDocCtrlEntry 2 }

**wwwDocCtrlBuckets OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The maximum number of buckets maintained by the agent before the oldest bucket is deleted. The buckets are used to populate the wwwDocAccessTopNTable and the wwwDocBytesTopNTable. The time interval captured in each bucket can be configured by setting the wwwDocCtrlBucketTimeInterval object."

DEFVAL { 4 } -- 4 buckets times 15 minutes = 1 hour

::= { wwwDocCtrlEntry 3 }



**wwwDocCtrlBucketTimeInterval OBJECT-TYPE**

SYNTAX TimeInterval

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The time interval after which a new bucket is created.

Changing this object has no effect on existing buckets."

DEFVAL { 90000 } -- 15 minutes (resolution .01 s)

::= { wwwDocCtrlEntry 4 }

**wwwDocCtrlTopNSize OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The maximum number of entries shown in the

wwwDocAccessTopNTable and the wwwDocBytesTopNTable.

Changing this object has no effect on existing buckets."

DEFVAL { 25 }

::= { wwwDocCtrlEntry 5 }

**wwwDocLastNTable OBJECT-TYPE**

SYNTAX SEQUENCE OF WwwDocLastNEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"The table which logs the last N access attempts."

::= { wwwDocumentStatistics 2 }

**wwwDocLastNEntry OBJECT-TYPE**

SYNTAX WwwDocLastNEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry which describes a recent access attempt."

INDEX { wwwServiceIndex, wwwDocLastNIndex }

::= { wwwDocLastNTable 1 }

WwwDocLastNEntry ::= SEQUENCE {

wwwDocLastNIndex Unsigned32,

wwwDocLastNName WwwDocName,

wwwDocLastNTimeStamp DateAndTime,

wwwDocLastNRequestType WwwRequestType,

wwwDocLastNResponseType WwwResponseType,

wwwDocLastNStatusMsg Utf8String,

wwwDocLastNBytes Unsigned32

}

**wwwDocLastNIndex OBJECT-TYPE**

SYNTAX           Unsigned32 (1..4294967295)  
MAX-ACCESS       not-accessible  
STATUS           current  
DESCRIPTION  
    "An arbitrary monotonically increasing integer number used  
    for indexing the wwwDocLastNTable. The first document  
    accessed appears in the table with this index value equal  
    to one. Each subsequent document is indexed with the next  
    sequential index value. The Nth document accessed will be  
    indexed by N. This table presents a sliding window of the  
    last wwwDocCtrlLastNSize documents accessed. Thus, entries  
    in this table will be indexed by N-wwwDocCtrlLastNSize  
    thru N if N > wwwDocCtrlLastNSize and 1 thru N if  
    N <= wwwDocCtrlLastNSize.

    The wwwDocCtrlLastNLock attribute can be used to lock  
    this table to allow the manager to read its contents."  
::= { wwwDocLastNEntry 1 }

wwwDocLastNName OBJECT-TYPE  
SYNTAX           WwwDocName  
MAX-ACCESS       read-only  
STATUS           current  
DESCRIPTION  
    "The name of the document for which access was attempted."  
::= { wwwDocLastNEntry 2 }

wwwDocLastNTimeStamp OBJECT-TYPE  
SYNTAX           DateAndTime  
MAX-ACCESS       read-only  
STATUS           current  
DESCRIPTION  
    "The date and time of the last attempt to access this  
    document."  
::= { wwwDocLastNEntry 3 }

wwwDocLastNRequestType OBJECT-TYPE  
SYNTAX           WwwRequestType  
MAX-ACCESS       read-only  
STATUS           current  
DESCRIPTION  
    "The protocol request type which was received by the  
    server when this document access was attempted."  
::= { wwwDocLastNEntry 4 }

wwwDocLastNResponseType OBJECT-TYPE  
SYNTAX           WwwResponseType  
MAX-ACCESS       read-only  
STATUS           current  
DESCRIPTION

"The protocol response type which was sent to the client as a result of this attempt to access a document. This object contains the type of the primary response if there were multiple responses to a single request."  
 ::= { wwwDocLastNEntry 5 }

wwwDocLastNStatusMsg OBJECT-TYPE

SYNTAX            Utf8String

MAX-ACCESS      read-only

STATUS           current

DESCRIPTION

"This object contains a human readable description of the reason why the wwwDocLastNResponseType was returned to the client. This object defines the implementation-specific reason if the value of wwwDocLastNResponseType indicates an error. For example, this object can indicate that the requested document could not be transferred due to a timeout condition or the document could not be transferred because a 'soft link' pointing to the document could not be resolved."

::= { wwwDocLastNEntry 6 }

wwwDocLastNBytes OBJECT-TYPE

SYNTAX           Unsigned32

MAX-ACCESS      read-only

STATUS           current

DESCRIPTION

"The number of content bytes that were returned as a result of this attempt to access a document."

::= { wwwDocLastNEntry 7 }

wwwDocBucketTable OBJECT-TYPE

SYNTAX           SEQUENCE OF WwwDocBucketEntry

MAX-ACCESS      not-accessible

STATUS           current

DESCRIPTION

"This table provides administrative summary information for the buckets maintained per WWW service."

::= { wwwDocumentStatistics 3 }

wwwDocBucketEntry OBJECT-TYPE

SYNTAX           WwwDocBucketEntry

MAX-ACCESS      not-accessible

STATUS           current

DESCRIPTION

"An entry which describes the parameters associated with a particular bucket."

INDEX           { wwwServiceIndex, wwwDocBucketIndex }

::= { wwwDocBucketTable 1 }

```
WwwDocBucketEntry ::= SEQUENCE {
    wwwDocBucketIndex      Unsigned32,
    wwwDocBucketTimeStamp  DateAndTime,
    wwwDocBucketAccesses   Unsigned32,
    wwwDocBucketDocuments  Unsigned32,
    wwwDocBucketBytes      Unsigned32
}

wwwDocBucketIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An arbitrary monotonically increasing integer number
         used for indexing the wwwDocBucketTable. The index number
         wraps to 1 whenever the maximum value is reached."
    ::= { wwwDocBucketEntry 1 }

wwwDocBucketTimeStamp OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The date and time when the bucket was made available."
    ::= { wwwDocBucketEntry 2 }

wwwDocBucketAccesses OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The total number of access attempts for any document
         provided by this WWW service during the time interval
         over which this bucket was created."
    ::= { wwwDocBucketEntry 3 }

wwwDocBucketDocuments OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The total number of different documents for which access
         was attempted this this WWW service during the time interval
         over which this bucket was created."
    ::= { wwwDocBucketEntry 4 }

wwwDocBucketBytes OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
```

## DESCRIPTION

"The total number of content bytes which were transferred from this WWW service during the time interval over which this bucket was created."

::= { wwwDocBucketEntry 5 }

## wwwDocAccessTopNTable OBJECT-TYPE

SYNTAX SEQUENCE OF WwwDocAccessTopNEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"The table of the most frequently accessed documents in a given bucket. This table is sorted by the column wwwDocAccessTopNAccesses. Entries having the same number of accesses are secondarily sorted by wwwDocAccessTopNBytes. Entries with the same number of accesses and the same number of bytes will have an arbitrary order."

::= { wwwDocumentStatistics 4 }

## wwwDocAccessTopNEntry OBJECT-TYPE

SYNTAX WwDocAccessTopNEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry in the top N table sorted by document accesses."

INDEX { wwwServiceIndex, wwwDocBucketIndex,  
wwwDocAccessTopNIndex }

::= { wwwDocAccessTopNTable 1 }

## WwDocAccessTopNEntry ::= SEQUENCE {

wwwDocAccessTopNIndex	Unsigned32,
wwwDocAccessTopNName	WwDocName,
wwwDocAccessTopNAccesses	Unsigned32,
wwwDocAccessTopNBytes	Unsigned32,
wwwDocAccessTopNLastResponseType	WwResponseType

}

## wwwDocAccessTopNIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An arbitrary monotonically increasing integer number used for indexing the wwwDocAccessTopNTable. The index is inversely correlated to the sorting order of the table. The document with the highest access count will get the index value 1."

::= { wwwDocAccessTopNEntry 1 }

**wwwDocAccessTopNName OBJECT-TYPE**

SYNTAX WwDocName

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The name of the document for which access was attempted."

::= { wwwDocAccessTopNEntry 2 }

**wwwDocAccessTopNAccesses OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of access attempts for this document."

::= { wwwDocAccessTopNEntry 3 }

**wwwDocAccessTopNBytes OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of content bytes that were transmitted  
as a result of attempts to access this document."

::= { wwwDocAccessTopNEntry 4 }

**wwwDocAccessTopNLastResponseType OBJECT-TYPE**

SYNTAX WwResponseType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The protocol response type which was sent to the client  
as a result of the last attempt to access this document.  
This object contains the type of the primary response if  
there were multiple responses to a single request."

::= { wwwDocAccessTopNEntry 5 }

**wwwDocBytesTopNTable OBJECT-TYPE**

SYNTAX SEQUENCE OF WwDocBytesTopNEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The table of the documents which caused most network  
traffic in a given bucket. This table is sorted by the  
column wwwDocBytesTopNBytes. Entries having the same number  
bytes are secondarily sorted by wwwDocBytesTopNAccesses.  
Entries with the same number of accesses and the same  
number of bytes will have an arbitrary order."

::= { wwwDocumentStatistics 5 }

```

wwwDocBytesTopNEntry OBJECT-TYPE
    SYNTAX      WwDocBytesTopNEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the top N table sorted by network traffic."
    INDEX       { wwwServiceIndex, wwwDocBucketIndex,
                  wwwDocBytesTopNIndex }
    ::= { wwwDocBytesTopNTable 1 }

WwDocBytesTopNEntry ::= SEQUENCE {
    wwwDocBytesTopNIndex      Unsigned32,
    wwwDocBytesTopNName      WwDocName,
    wwwDocBytesTopNAccesses  Unsigned32,
    wwwDocBytesTopNBytes     Unsigned32,
    wwwDocBytesTopNLastResponseType WwResponseType
}

wwwDocBytesTopNIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An arbitrary monotonically increasing integer number
        used for indexing the wwwDocBytesTopNTable. The index is
        inversely correlated to the sorting order of the table. The
        document with the highest byte count will get the index
        value 1."
    ::= { wwwDocBytesTopNEntry 1 }

wwwDocBytesTopNName OBJECT-TYPE
    SYNTAX      WwDocName
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The name of the document for which access was attempted."
    ::= { wwwDocBytesTopNEntry 2 }

wwwDocBytesTopNAccesses OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of access attempts for this document."
    ::= { wwwDocBytesTopNEntry 3 }

wwwDocBytesTopNBytes OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current

```

## DESCRIPTION

"The total number of content bytes that were transmitted as a result of attempts to access this document."

::= { wwwDocBytesTopNEntry 4 }

## wwwDocBytesTopNLastResponseType OBJECT-TYPE

SYNTAX WwvResponseType

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The protocol response type which was sent to the client as a result of the last attempt to access this document. This object contains the type of the primary response if there were multiple responses to a single request."

::= { wwwDocBytesTopNEntry 5 }

--

-- Conformance Definitions

--

wwwMIBCompliances OBJECT IDENTIFIER ::= { wwwMIBConformance 1 }

wwwMIBGroups OBJECT IDENTIFIER ::= { wwwMIBConformance 2 }

## wwwMinimalCompliance MODULE-COMPLIANCE

STATUS current

## DESCRIPTION

"The compliance statement for SNMP agents which implement the minimal subset of the WWW-MIB. Implementors might choose this subset for high-performance server where full compliance might be to expensive."

MODULE -- this module

MANDATORY-GROUPS {  
    wwwServiceGroup,  
    wwwSummaryGroup  
}

OBJECT wwwSummaryOutRequests

## DESCRIPTION

"Instances of wwwSummaryOutRequests do not exist on pure WWW server implementations."

OBJECT wwwSummaryInResponses

## DESCRIPTION

"Instances of wwwSummaryOutRequests do not exist on pure WWW server implementations."

OBJECT wwwSummaryInRequests

## DESCRIPTION

"Instances of wwwSummaryInRequests do not exist on pure WWW client implementations."

OBJECT wwwSummaryOutResponses

## DESCRIPTION

"Instances of wwwSummaryOutResponses do not exist on pure



```
        WWW client implementations."
 ::= { wwwMIBCompliances 1 }

wwwFullCompliance MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "The compliance statement for SNMP agents which implement
         the full WWW-MIB."
    MODULE -- this module
    MANDATORY-GROUPS {
        wwwServiceGroup,
        wwwSummaryGroup
    }
    GROUP wwwRequestInGroup
    DESCRIPTION
        "The wwwRequestInGroup is mandatory only for WWW server
         or proxy server implementations."
    GROUP wwwResponseOutGroup
    DESCRIPTION
        "The wwwResponseOutGroup is mandatory only for WWW server
         or proxy server implementations."
    GROUP wwwRequestOutGroup
    DESCRIPTION
        "The wwwRequestOutGroup is mandatory only for WWW client
         or proxy server implementations."
    GROUP wwwResponseInGroup
    DESCRIPTION
        "The wwwRequestOutGroup is mandatory only for WWW client
         or proxy server implementations."
    GROUP wwwDocumentGroup
    DESCRIPTION
        "The wwwDocumentGroup is mandatory only for WWW server
         or proxy server implementations."
    OBJECT wwwSummaryOutRequests
    DESCRIPTION
        "Instances of wwwSummaryOutRequests do not exist on pure
         WWW server implementations."
    OBJECT wwwSummaryInResponses
    DESCRIPTION
        "Instances of wwwSummaryOutRequests do not exist on pure
         WWW server implementations."
    OBJECT wwwSummaryInRequests
    DESCRIPTION
        "Instances of wwwSummaryInRequests do not exist on pure
         WWW client implementations."
    OBJECT wwwSummaryOutResponses
    DESCRIPTION
        "Instances of wwwSummaryOutResponses do not exist on pure
         WWW client implementations."
 ::= { wwwMIBCompliances 2 }
```

```
wwwServiceGroup OBJECT-GROUP
  OBJECTS {
    wwwServiceDescription,
    wwwServiceContact,
    wwwServiceProtocol,
    wwwServiceName,
    wwwServiceType,
    wwwServiceStartTime,
    wwwServiceOperStatus,
    wwwServiceLastChange
  }
  STATUS      current
  DESCRIPTION
    "A collection of objects providing information about
    the WWW services known by the SNMP agent."
  ::= { wwwMIBGroups 1 }

wwwSummaryGroup OBJECT-GROUP
  OBJECTS {
    wwwSummaryInRequests,
    wwwSummaryOutRequests,
    wwwSummaryInResponses,
    wwwSummaryOutResponses,
    wwwSummaryInBytes,
    wwwSummaryInLowBytes,
    wwwSummaryOutBytes,
    wwwSummaryOutLowBytes
  }
  STATUS      current
  DESCRIPTION
    "A collection of objects providing summary statistics
    about requests and responses generated and received
    by a WWW service."
  ::= { wwwMIBGroups 2 }

wwwRequestInGroup OBJECT-GROUP
  OBJECTS {
    wwwRequestInRequests,
    wwwRequestInBytes,
    wwwRequestInLastTime
  }
  STATUS      current
  DESCRIPTION
    "A collection of objects providing detailed statistics
    about requests received by a WWW service."
  ::= { wwwMIBGroups 3 }

wwwRequestOutGroup OBJECT-GROUP
  OBJECTS {
    wwwRequestOutRequests,
```

```
        wwwRequestOutBytes,
        wwwRequestOutLastTime
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing detailed statistics
        about requests generated by a WWW service."
    ::= { wwwMIBGroups 4 }

wwwResponseInGroup OBJECT-GROUP
    OBJECTS {
        wwwResponseInResponses,
        wwwResponseInBytes,
        wwwResponseInLastTime
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing detailed statistics
        about responses received by a WWW service."
    ::= { wwwMIBGroups 5 }

wwwResponseOutGroup OBJECT-GROUP
    OBJECTS {
        wwwResponseOutResponses,
        wwwResponseOutBytes,
        wwwResponseOutLastTime
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing detailed statistics
        about responses generated by a WWW service."
    ::= { wwwMIBGroups 6 }

wwwDocumentGroup OBJECT-GROUP
    OBJECTS {
        wwwDocCtrlLastNSize,
        wwwDocCtrlLastNLock,
        wwwDocCtrlBuckets,
        wwwDocCtrlBucketTimeInterval,
        wwwDocCtrlTopNSize,
        wwwDocLastNName,
        wwwDocLastNTimeStamp,
        wwwDocLastNRequestType,
        wwwDocLastNResponseType,
        wwwDocLastNStatusMsg,
        wwwDocLastNBytes,
        wwwDocBucketTimeStamp,
        wwwDocBucketAccesses,
        wwwDocBucketDocuments,
        wwwDocBucketBytes,
    }
```

```

        wwwDocAccessTopNName,
        wwwDocAccessTopNAccesses,
        wwwDocAccessTopNBytes,
        wwwDocAccessTopNLastResponseType,
        wwwDocBytesTopNName,
        wwwDocBytesTopNAccesses,
        wwwDocBytesTopNBytes,
        wwwDocBytesTopNLastResponseType
    }
    STATUS    current
    DESCRIPTION
        "A collection of objects providing information about
        accesses to documents."
    ::= { wwwMIBGroups 7 }

END

```

## 7. Document Transfer Protocol Mappings

This section describes how existing protocols such as HTTP [19,20] and FTP [21] can be mapped on the abstract Document Transfer Protocol (DTP) used within the definitions of the WWW MIB. Every mapping must define the identifier which is used to uniquely identify the transfer protocol. In addition, the mappings must define how requests and responses are identified.

### 7.1. The HyperText Transfer Protocol

The HyperText Transfer Protocol (HTTP) [19,20] is an application-level protocol used to transfer hypermedia documents in a distributed networked environment. HTTP is based on the request/response paradigm and can be mapped on the abstract DTP easily.

The HTTP protocol usually runs over TCP and uses the well-known TCP port 80. Therefore, the default value for the `wwwServiceProtocol` object is { `applTCPProtoID 80` }.

HTTP allows for both requests and responses and an open-ended set of message types. The general message syntax of HTTP is therefore used for the protocol mapping. The BNF specification of the general HTTP message syntax as defined in [20] is as follows:

```

generic-message = start-line
                  *message-header
                  CRLF
                  [ message-body ]

start-line      = Request-Line | Status-Line

```

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

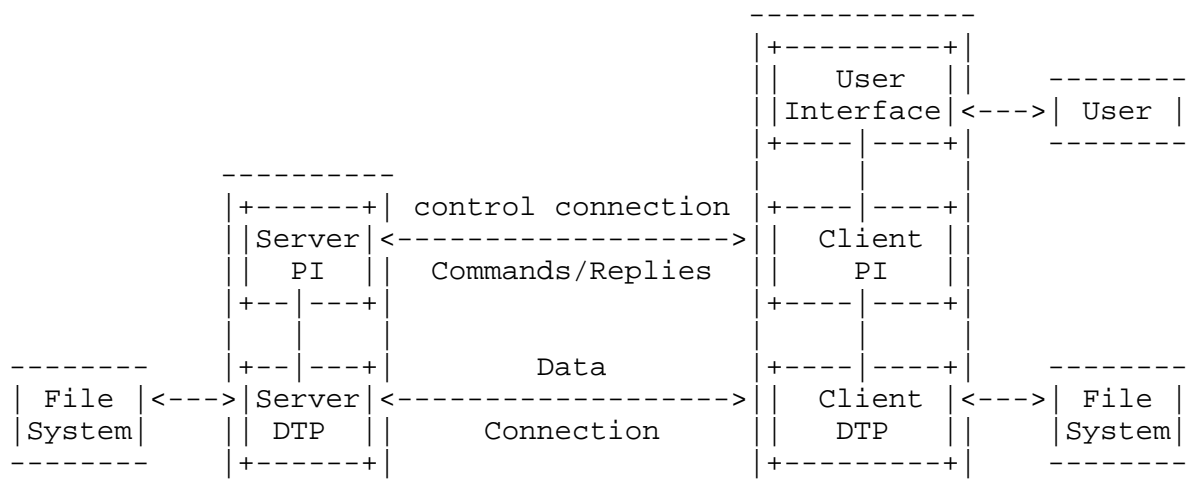
Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

Every HTTP-message where the start-line is a Request-Line is considered a request in the abstract DTP. Every HTTP-message where the start-line is a Status-Line is considered a response in the abstract DTP. The mappings of WwwRequestType and WwwResponseType are defined as follows:

- o The WwwRequestType corresponds to the method token in the Request-Line.
- o The WwwResponseType corresponds to the Status-Code in the Status-Line.

## 7.2. The File Transfer Protocol

The File Transfer Protocol (FTP) [21] is an application-level protocol used to transfer files between hosts connected by the TCP/IP suite of protocols. FTP is based on a request/response paradigm and is mapped on the abstract DTP as defined in this section. The FTP model as defined in [21] is depicted below.



FTP uses two different connection types between a client and a server to transfer files. The control connection is persistent during a FTP session and used to exchange FTP commands and associated replies. The data connection is only available when bulk data has to be transferred.

The FTP protocol usually runs over TCP and uses the well-known TCP port 21 to setup the control connection. Therefore, the default value

for the `wwwServiceProtocol` object is { `applTCPProtoID 21` }.

Every FTP command is considered a request in the abstract DTP. Every FTP reply is considered a response in the abstract DTP. It should be noted that a single FTP command can result in multiple FTP replies (e.g. preliminary positive replies). The primary response for a FTP request contains a status code of the form `2xy`, `3xy`, `4xy` or `5xy`. See section 4.2 in [21] for the exact meaning of these status codes. The mappings for `WwwRequestType` and `WwwResponseType` are defined as follows:

- o The `WwwRequestType` corresponds to the FTP command token.
- o The `WwwResponseType` corresponds to the three-digit code which starts a reply. Multi-line replies with the same three-digit code are counted as a single DTP response.

## 8. Security Considerations

There are a number of management objects defined in this MIB module that have a `MAX-ACCESS` clause of `read-write`. Such objects may be considered sensitive or vulnerable in some network environments. The support for write operations in a non-secure environment without proper protection can have a negative effect on network operations.

There are a number of managed objects in this MIB that may contain sensitive information:

- o The document statistics group contains traffic information including the names of documents that were a target of protocol operations. This information is sensitive as it allows to obtain access statistics for documents.
- o The protocol statistics are less sensitive, because they do not contain details about the target of individual requests and responses. However, traffic statistics and error counters still provide usage information about WWW services and about the overall quality of WWW services. It is suggested that sites configure MIB views so that a user of this MIB can only access the portion of the statistics that belong to the WWW services managed by that user.
- o The service and the summary statistics groups provide information about the existence of WWW services and condensed usage statistics. Some sites may want to protect this information as well, especially if they offer private WWW services that should not be known by the outside world.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), there is no control as to who on the secure network is allowed to access (read/change/create/delete) the objects in this MIB.

It is recommended that implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [12] and the View-based Access Control Model RFC 2575 [15] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed read or write (change/create/delete) them.

## 9. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

## 10. Acknowledgments

This document was produced by the Application MIB working group. The editors gratefully acknowledge the comments of the following individuals:

Mark Gamble, Cheryl Krupczak, Randy Presuhn, Jon Saperia,  
Bob Stewart, Martin Toet, Chris Wellens, Kenneth White.

## 11. Editors' Addresses

Harrie Hazewinkel  
Joint Research Centre of the E.C.  
via Fermi - Ispra 21020 (VA)  
Italy

Phone: +39 0332786322  
Fax: +39 0332785641  
EMail: harrie.hazewinkel@jrc.it

Carl W. Kalbfleisch  
Verio, Inc.  
1950 Stemmons Frwy  
Suite 2006  
Dallas, TX 75207  
USA

Phone: +1 214 290-8653  
Fax: +1 214 744-0742  
EMail: cwk@verio.net

Juergen Schoenwaelder  
TU Braunschweig  
Bueltenweg 74/75  
38106 Braunschweig  
Germany

Phone: +49 531 391-3683  
Fax: +49 531 489-5936  
EMail: schoenw@ibr.cs.tu-bs.de

## 12. References

- [1] Wijnen,, B., Harrington, D. and R. Presuhn, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD, 16, RFC 1155, May 1990.
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, Performance Systems International, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.



- [5] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIV2)", STD 58, RFC 2578, April 1999.
- [6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIV2", STD 58, RFC 2579, April 1999.
- [7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIV2", STD 58, RFC 2580, April 1999.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.
- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.
- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.
- [13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC 2573, April 1999.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.
- [16] Hovey, R. and S. Bradner, "The Organizations Involved in the IETF Standards Process", BCP 11, RFC 2028, October 1996.

- [17] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [18] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.
- [19] Berners-Lee, T., Fielding, R. and H. Frystyk, "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996.
- [20] Fielding, R., Gettys, J., Mogul, J., Frystyk, H. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.
- [21] Postel, J. and J. Reynolds, "File Transfer Protocol (FTP)", STD 9, RFC 959, October 1985.
- [22] Kalbfleisch, C., "Applicability of Standards Track MIBs to Management of World Wide Web Servers", RFC 2039, November 1996.
- [23] Krupczak, C. and J. Saperia, "Definitions of System-Level Managed Objects for Applications", RFC 2287, February 1998.
- [24] Kalbfleisch, C., Krupczak, C., Presuhn, R. and J. Saperia, "Application Management MIB", RFC 2564, May 1999.
- [25] Kantor, B. and P. Lapsley, "Network News Transfer Protocol: A Proposed Standard for the Stream-Based Transmission of News", RFC 977, February 1986.
- [26] Callaghan, B., "WebNFS Client Specification", RFC 2054, October 1996
- [27] Callaghan, B., "WebNFS Server Specification", RFC 2055, October 1996.

### 13. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

### Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.