

Application Performance Measurement MIB

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 (2004). All Rights Reserved.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for measuring the application performance as experienced by end-users.

Table of Contents

1.	The Internet-Standard Management Framework	2
2.	Overview	2
2.1.	Report Aggregation	4
2.2.	AppLocalIndex Linkages	8
2.3.	Measurement Methodology.	10
2.4.	Instrumentation Architectures.	10
2.4.1.	Application Directory Caching.	10
2.4.2.	Push Model	11
2.5.	Structure of this MIB Module	12
2.5.1.	The APM Application Directory Group.	13
2.5.2.	The APM User Defined Applications Group.	13
2.5.3.	The APM Report Group	13
2.5.4.	The APM Transaction Group.	13
2.5.5.	The APM Exception Group.	14
2.5.6.	The APM Notification Group	14
3.	Definitions.	14
4.	Security Considerations.	58
5.	References	60
5.1.	Normative References	60
5.2.	Informative References	60

6. Author's Address	60
7. Full Copyright Statement	61

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [8].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [1], STD 58, RFC 2579 [2] and STD 58, RFC 2580 [3].

2. Overview

This document continues the architecture created in the RMON MIB [7] by providing analysis of application performance as experienced by end-users.

Application performance measurement measures the quality of service delivered to end-users by applications. With this perspective, a true end-to-end view of the IT infrastructure results, combining the performance of the application, desktop, network, and server, as well as any positive or negative interactions between these components.

Despite all the technically sophisticated ways in which networking and system resources can be measured, human end-users perceive only two things about an application: availability and responsiveness.

Availability - The percentage of the time that the application is ready to give a user service.

Responsiveness - The speed at which the application delivers the requested service.

A transaction is an action initiated by a user that starts and completes a distributed processing function. A transaction begins when a user initiates a request for service (i.e., pushing a submit button) and ends when the work is completed (i.e., information is provided or a confirmation is delivered). A transaction is the fundamental item measured by the APM MIB.

A failed transaction is a transaction that fails to provide the service requested by the end user, regardless of whether it is due to a processing failure or transport failure.

An application protocol (e.g., POP3) may implement different commands or application "verbs" (e.g., POP3 Login and POP3 Retrieval). It will often be interesting to monitor these verbs separately because:

- 1) The verbs may have widely differing performance characteristics (in fact some may be response time oriented while others are throughput oriented)
- 2) The verbs have varying business significance
- 3) It provides more granularity of exactly what might be performing poorly

This MIB Module allows the measurement of a parent application, its component verbs, or both. If monitoring both, one can watch the top-level application and then drill down to the verbs when trouble is spotted to learn which subcomponents are in trouble. Each application verb is registered separately in the Protocol Directory [5] [6] as a child of its parent application.

Application protocols implement one of three different types of transactions: transaction-oriented, throughput-oriented, or streaming-oriented. While the availability metric is the same for all three types, the responsiveness metric varies:

Transaction-Oriented: These transactions have a fairly constant workload to perform for all transactions. In particular, to the degree that the workload may vary, it doesn't vary based on the amount of data to be transferred but based on the parameters of the transaction. The responsiveness metric for transaction-oriented applications is application response time, the elapsed time between the user's request for service (e.g., pushing the submit button) and the completion of the request (e.g., displaying the results) and is measured in milliseconds. This is commonly referred to as end-user response time.

Throughput-Oriented: These transactions have widely varying workloads based on the amount of data requested. The responsiveness metric for throughput-oriented applications is kilobits per second.

Streaming-Oriented: These transactions deliver data at a constant metered rate of speed regardless of excess capacity in the networking and computing infrastructure. However, when the infrastructures cannot deliver data at this speed, interruption of service or degradation of service can result. The responsiveness

metric for streaming-oriented applications is the signal quality ratio of time that the service is degraded or interrupted to the total service time. This metric is measured in parts per million.

2.1. Report Aggregation

This MIB Module provides functions to aggregate measurements into higher level summaries.

Every transaction is identified by its application, server, and client and has an availability measure as well as a responsiveness measure. The appropriate responsiveness measure is context-sensitive depending on whether the application is transaction-oriented, throughput-oriented, or streaming-oriented. For example, in a 5 minute period several transactions might be recorded:

Application	Client	Server	Successful	Responsiveness
HTTP	Jim	Sales	1	6 sec.
SAP/R3	Jane	Finance	1	17 sec.
HTTP	Joe	HR	0	-
FTP	Jim	FTP	1	212 Kbps
HTTP	Joe	HR	1	25 sec.
RealVideo	Joe	Videoconf	1	100.0%
HTTP	Jane	HR	1	5 sec.

These transactions can be aggregated in several ways, providing statistical summaries - for example summarizing all HTTP transactions, or all HTTP transactions to the HR Server. Note that data from different applications may not be summarized because:

1. The performance characteristics of different applications differ widely enough to render statistical analysis meaningless.
2. The responsiveness metrics of different applications may be different, making a statistical analysis impossible (in other words, one application may be transaction-oriented, while another is throughput-oriented).

Aggregating transactions collected over a period requires an aggregation algorithm. In this MIB Module, transaction aggregation always results in the following statistics:

TransactionCount

The total number of transactions during this period

SuccessfulTransactions

The total number of transactions that were successful. The management station can derive the percent success by dividing SuccessfulTransactions by the TransactionCount.

ResponsivenessMean

The average of the responsiveness metric for all aggregated transactions that completed successfully.

ResponsivenessMin

The minimum responsiveness metric for all aggregated transactions that completed successfully.

ResponsivenessMax

The maximum responsiveness metric for all aggregated transactions that completed successfully.

ResponsivenessBx

The count of successful transactions whose responsiveness metric fell into the range specified for Bx. There are 7 buckets specified. Because the performance of different applications varies widely, the bucket ranges are specified separately for each application (in the apmAppDirTable) so that they may be tuned to typical performance of each application.

For example, when aggregating the previous set of transactions by application we get (for simplicity the example only shows TransactionCount, SuccessfulTransactions, and ResponsivenessMean):

Application	Count	Successful	ResponsivenessMean
HTTP	4	3	12 sec.
SAP/R3	1	1	17 sec.
FTP	1	1	212 Kbps.
RealVideo	1	1	100.0%

There are four different types of aggregation.

The flows(1) aggregation is the simplest. All transactions that share common application/server/client 3-tuples are aggregated together, resulting in a set of metrics for all such unique 3-tuples.

The clients(2) aggregation results in somewhat more aggregation (i.e., fewer resulting records). All transactions that share common application/client tuples are aggregated together, resulting in a set of metrics for all such unique tuples.

The servers(3) aggregation usually results in still more aggregation (i.e., fewer resulting records). All transactions that share common application/server tuples are aggregated together, resulting in a set of metrics for all such unique tuples.

The applications(4) aggregation results in the most aggregation (i.e., the fewest resulting records). All transactions that share a common application are aggregated together, resulting in a set of metrics for all such unique applications.

For example, if in a 5 minute period the following transactions occurred:

Actual Transactions:

#	App	Client	Server	Successful	Responsiveness
1	HTTP	Jim	CallCtr	N	-
2	HTTP	Jim	HR	Y	12 sec.
3	HTTP	Jim	Sales	Y	7 sec.
4	HTTP	Jim	CallCtr	Y	5 sec.
5	Email	Jim	Pop3	Y	12 sec.
6	HTTP	Jane	CallCtr	Y	3 sec.
7	SAP/R3	Jane	Finance	Y	19 sec.
8	Email	Jane	Pop3	Y	16 sec.
9	HTTP	Joe	HR	Y	18 sec.

The flows(1) aggregation results in the following table. Note that the first record (HTTP/Jim/CallCtr) is the aggregation of transactions #1 and #4:

Flow Aggregation:

App	Client	Server	Count	Succe- ssful	Rsp Mean	Rsp Min	Rsp Max	RspB1	RspB2
HTTP	Jim	CallCtr	2	1	5	5	5	1	0
HTTP	Jim	HR	1	1	12	12	12	0	1
HTTP	Jim	Sales	1	1	7	7	7	1	0
Email	Jim	Pop3	1	1	12	12	12	0	1
HTTP	Jane	CallCtr	1	1	3	3	3	1	0
SAP/R3	Jane	Finance	1	1	19	19	19	0	1
Email	Jane	Pop3	1	1	16	16	16	0	1
HTTP	Joe	HR	1	1	18	18	18	0	1

(Note: Columns above such as RspMean and RspB1 are abbreviations for objects in the apmReportTable)

The clients(2) aggregation results in the following table. Note that the first record (HTTP/Jim) is the aggregate of transactions #1, #2, #3 and #4:

Client Aggregation:

App	Client	Count	Succe- ssful	Rsp Mean	Rsp Min	Rsp Max	RspB1	RspB2 ...
HTTP	Jim	4	3	8	5	12	2	1
Email	Jim	1	1	12	12	12	0	1
HTTP	Jane	1	1	3	3	3	1	0
SAP/R3	Jane	1	1	19	19	19	0	1
Email	Jane	1	1	16	16	16	0	1
HTTP	Joe	1	1	18	18	18	0	1

The servers(3) aggregation results in the following table. Note that the first record (HTTP/CallCtr) is the aggregation of transactions #1, #4 and #6:

Server Aggregation:

App	Server	Count	Succe- ssful	Rsp Mean	Rsp Min	Rsp Max	RspB1	RspB2 ...
HTTP	CallCtr	3	2	4	3	5	2	0
HTTP	HR	2	2	15	12	18	0	2
HTTP	Sales	1	1	7	7	7	1	0
Email	Pop3	2	2	14	12	16	0	2
SAP/R3	Finance	1	1	19	19	19	0	1

The applications(4) aggregation results in the following table. Note that the first record (HTTP) is the aggregate of transactions #1, #2, #3, #5, #6 and #9:

Application Aggregation:

App	Count	Succe- ssful	Rsp Mean	Rsp Min	Rsp Max	RspB1	RspB2 ...
HTTP	6	5	9	3	18	3	2
Email	2	2	14	12	16	0	2
SAP/R3	1	1	19	19	19	0	1

The apmReportControlTable provides for a historical set of the last 'X' reports, combining the historical records found in history tables with the periodic snapshots found in TopN tables. Conceptually the components are:

apmReportControlTable

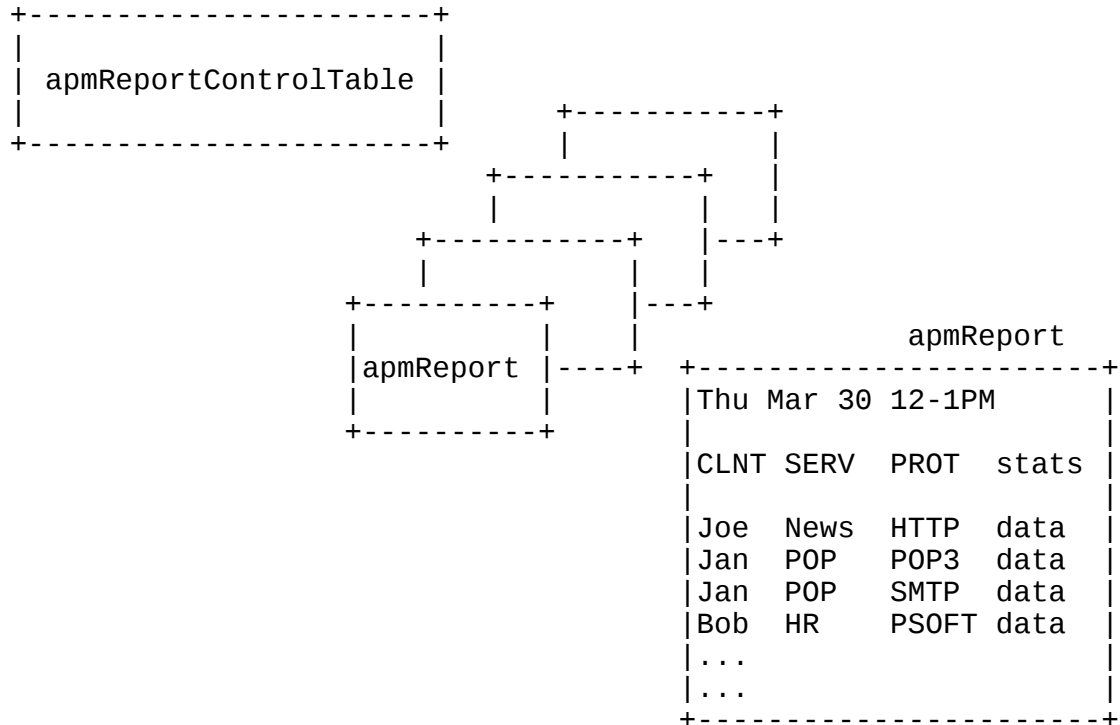
Specifies data collection and summarization parameters, including the number of reports to keep and the size of each report.

apmReport

Each APM Report contains an aggregated list of records that represent data collected during a specific time period.

An `apmReportControlEntry` causes a family of APM Reports to be created, where each report summarizes different, successive, contiguous periods of time.

While the conceptual model of APM Reports shows them as distinct entities, they are all entries in a single `apmReportTable`, where entries in report 'A' are separated from entries in report 'B' by different values of the `apmReportIndex`.



2.2. AppLocalIndex Linkages

The following set of example tables illustrates a few points:

1. How `protocolDirEntries`, `apmHttpFilterEntries` and `apmUserDefinedAppEntries`(not shown) all result in entries in the `apmAppDirTable`.
2. How a single `appLocalIndex` may be represented multiple times in the `apmAppDirTable` and `apmReportTable` if the agent measures multiple responsiveness types for that application.

A convention in the formatting of these tables is that the columns to the left of the '|' separator are index columns for the table.

Assuming the following entries in the `RMON2 protocolDirectory`:

protocolDirectory		
ID (*)	Parameters	LocalIndex ...
WWW	None	1
WWW Get	None	2
SAP/R3	None	3

(*) These IDs are represented here symbolically. Consult [5] for more detail in their format

and the following entry in the `apmHttpFilterTable`:

ApmHttpFilterTable				
Index	AppLocalIndex	ServerAddress	URLPath	MatchType ...
5	20	hr.example.com	/expense	prefix(3) ...

the `apmAppDirTable` would be populated with the following entries:

apmAppDir		
AppLocalIndex	ResponsivenessType	Config ...
1	transaction(1)	On ...
1	throughput(2)	On ...
2	transaction(1)	On ...
2	throughput(2)	On ...
3	transaction(1)	On ...
20	transaction(1)	On ...
20	throughput(2)	On ...

The entries in the `apmAppDirTable` with an `appLocalIndex` of 1, 2 and 3 correspond to the identically named entries in the `protocolDirectory` table. `appLocalIndex` #1 results in 2 entries, one to measure the transaction responsiveness of WWW and one to measure its throughput responsiveness. In contrast, `appLocalIndex` #3 results in only a transaction entry because the agent does not measure the throughput responsiveness for SAP/R3 (probably because it isn't very meaningful). Finally, `appLocalIndex` #20 corresponds to the entry in the `apmHttpFilterTable` and has transaction responsiveness and throughput responsiveness measurements available.

If a report was configured using application aggregation, entries in that report might look like:

apmReportTable				
CtlIndex	Index	AppLocalIdx	ResponsivenessType	TransactionCount ...
1	1	1	transaction(1)	counters...
1	1	1	throughput(2)	counters...
1	1	2	transaction(1)	counters...
1	1	2	throughput(2)	counters...
1	1	3	transaction(1)	counters...
1	1	20	transaction(1)	counters...
1	1	20	throughput(2)	counters...

Note that the index items `protocolDirLocalIndex`, `apmReportServerAddress` and `apmReportClientID` were omitted from `apmReportTable` example for brevity because they would have been equal to zero due to the use of the application aggregation in this example.

2.3. Measurement Methodology

There are many different measurement methodologies available for measuring application performance (e.g., probe-based, client-based, synthetic-transaction, etc.). This specification does not mandate a particular methodology - it is open to any that meet the minimum requirements. Conformance to this specification requires that the collected data match the semantics described herein. In particular, a data collection methodology must be able to measure response time, throughput, streaming responsiveness and availability as specified.

Note that in some cases a transaction may run for a long time but ultimately be successful. The measurement software shouldn't prematurely classify lengthy transactions as failures but should wait as long as the client application will wait for a successful response.

2.4. Instrumentation Architectures

Different architectural approaches and deployment strategies may be taken towards implementation of this specification. If a highly distributed approach is desired (e.g., an agent per desktop), one or both of the two approaches below may be used to make it more practical.

2.4.1. Application Directory Caching

It is necessary for the manager to have a copy of the tables that define the Application Directory in order to interpret APM measurements. It is likely that in a highly distributed network of

thousands of APM agents, this Application Directory will be the same on many, if not all of the agents. Repeated downloads of the Application Directory may be inefficient.

The `apmAppDirID` object is a single object that identifies the configuration of all aspects of the Application Directory when it is equal to a well-known, registered configuration. Thus, when a manager sees an `apmAppDirID` value that it recognizes, it need not download the Application Directory from that agent. In fact, the manager may discover a new registered Application Directory configuration on one agent and then re-use that configuration on another agent that shares the same `apmAppDirID` value.

Application directory registrations are unique within an administrative domain, allowing an administrator to create a custom application directory configuration without the need to assign it a globally-unique registration.

2.4.2. Push Model

When APM agents are installed on "desktops" (including laptops), a few issues make polling difficult:

1. Desktops often have dynamically-assigned addresses so there is no long-lived address to poll.
2. Desktops are not available as much as infrastructure components due to crashes, user-initiated reboots and shutdowns and user control over monitoring software. Thus a desktop may not be available to answer a poll at the moment when the manager is scheduled to poll that desktop.
3. Laptops that are connected via dialup connections are only sporadically connected and will routinely be unreachable when the manager is scheduled to poll.

As a consequence, a push model is usually more appropriate for desktop-based agents. To achieve this, the agent should follow the following rules in deciding what data to send in notifications.

APM Reports

If an agent wishes to push APM reports to a manager, it must send:

apmAppDirID

apmNameTable (any data updated since the last push)

For each report the agent wishes to upload, it must send the entire apmReportControlEntry associated with that report and the associated entries in the apmReportTable that have changed since the last report.

APM Transactions

If an agent wishes to push APM transactions to a manager, it must send:

apmAppDirID

apmNameTable (any data updated since the last push)

apmTransactionTable (relevant entries)

APM Exceptions

The agent must send:

apmAppDirID

apmNameTable (any data updated since the last push)

apmTransactionEntry (of exception transaction)

apmExceptionEntry (entry that generated exception)

[Note that this list supersedes the information in the OBJECTS clauses of the apmTransactionResponsivenessAlarm and apmTransactionUnsuccessfulAlarm when the agent is using a push model. This additional information eliminates the need for the manager to request additional data to understand the exception.]

The order of varbinds and where to segment varbinds into PDUs is at the discretion of the agent.

2.5. Structure of this MIB Module

The objects are arranged into the following groups:

- APM Application Directory Group
- APM User Defined Applications Group
- APM Report Group
- APM Transaction Group
- APM Exception Group
- APM Notification Group

These groups are the basic unit of conformance. If an agent implements a group, then it must implement all objects in that group. While this section provides an overview of grouping and conformance information for this MIB Module, the authoritative reference for such information is contained in the MODULE-COMPLIANCE and OBJECT-GROUP macros later in this MIB Module.

These groups are defined to provide a means of assigning object identifiers, and to provide a method for implementors of managed agents to know which objects they must implement.

2.5.1. The APM Application Directory Group

The APM Application Directory group contains configuration objects for every application or application verb monitored on this system. This group consists of the `apmAppDirTable`.

2.5.2. The APM User Defined Applications Group

The APM User Defined Applications Group contains objects that allow for the tracking of applications or application verbs that aren't registered in the `protocolDirTable`. This group consists of the `apmHttpFilterTable` and the `apmUserDefinedAppTable`.

2.5.3. The APM Report Group

The APM Report Group is used to prepare regular reports that aggregate application performance by flow, by client, by server, or by application. This group consists of the `apmReportControlTable` and the `apmReportTable`.

2.5.4. The APM Transaction Group

The APM Transaction Group is used to show transactions that are currently in progress and ones that have ended recently, along with their responsiveness metric.

Because many transactions last a very short time and because an agent may not retain completed transactions very long, transactions may exist in this table for a very short time. Thus, polling this table isn't an effective mechanism for retrieving all transactions unless the value of `apmTransactionsHistorySize` is suitably large for the transactions being monitored.

One important benefit of this table is that it allows a management station to check on the status of long-lived transactions. Because the `apmReport` and `apmException` mechanisms act only on transactions that have finished, a network manager may not have visibility for

some time into the performance of long-lived transactions such as streaming applications, large data transfers, or (very) poorly performing transactions. In fact, by their very definition, the `apmReport` and `apmException` mechanisms only provide visibility into a problem after nothing can be done about it. This group consists primarily of the `apmTransactionTable`.

2.5.5. The APM Exception Group

The APM Exception Group is used to generate immediate notifications of transactions that cross certain thresholds. The `apmExceptionTable` is used to configure which thresholds are to be checked for which types of transactions. The `apmTransactionResponsivenessAlarm` notification is sent when a transaction occurs with a responsiveness that crosses a threshold. The `apmTransactionUnsuccessfulAlarm` notification is sent when a transaction fails for which exception checking was configured. This group consists primarily of the `apmExceptionTable`.

2.5.6. The APM Notification Group

The APM Notification Group contains 2 notifications that are sent when thresholds in the APM Exception Table are exceeded.

3. Definitions

APM-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY, OBJECT-TYPE,
NOTIFICATION-TYPE,
Counter32, Unsigned32

FROM SNMPv2-SMI

TEXTUAL-CONVENTION, RowStatus, TimeStamp,
TimeInterval, TruthValue, DateAndTime,
StorageType

FROM SNMPv2-TC

MODULE-COMPLIANCE, OBJECT-GROUP,
NOTIFICATION-GROUP

FROM SNMPv2-CONF

SnmpAdminString

FROM SNMP-FRAMEWORK-MIB

rmon, OwnerString

FROM RMON-MIB

protocolDirLocalIndex

FROM RMON2-MIB;

-- Application Performance Measurement MIB

apm MODULE-IDENTITY

LAST-UPDATED "200402190000Z" -- February 19, 2004

ORGANIZATION "IETF RMON MIB Working Group"

CONTACT-INFO

"Author:

Steve Waldbusser

Phone: +1-650-948-6500
 Fax : +1-650-745-0671
 Email: waldbusser@nextbeacon.com

Working Group Chair:
 Andy Bierman
 Cisco Systems, Inc.
 Postal: 170 West Tasman Drive
 San Jose, CA USA 95134
 Tel: +1 408 527-3711
 E-mail: abierman@cisco.com

Working Group Mailing List: <rmonmib@ietf.org>
 To subscribe send email to: <rmonmib-request@ietf.org>

"

DESCRIPTION

"The MIB module for measuring application performance as experienced by end-users.

Copyright (C) The Internet Society (2004). This version of this MIB module is part of RFC 3729; see the RFC itself for full legal notices."

REVISION "200402190000Z" -- February 19, 2004

DESCRIPTION

"The original version of this MIB Module, published as RFC3729."

::= { rmon 23 }

apmMibObjects OBJECT IDENTIFIER ::= { apm 1 }
 apmConformance OBJECT IDENTIFIER ::= { apm 2 }
 apmCompliances OBJECT IDENTIFIER ::= { apmConformance 1 }
 apmGroups OBJECT IDENTIFIER ::= { apmConformance 2 }

AppLocalIndex ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A locally arbitrary unique identifier associated with an application or application verb.

All objects of type AppLocalIndex are assigned by the agent out of a common number space. In other words, AppLocalIndex values assigned to entries in one table must not overlap with AppLocalIndex values assigned to entries in another table. Further, every protocolDirLocalIndex value registered by the agent automatically assigns the same value out of the

AppLocalIndex number space.

For example, if the protocolDirLocalIndex values { 1, 3, 5, 7 } have been assigned, and the apmHttpFilterAppLocalIndex values { 6, 8, 9 } have been assigned:

- Assignment of new AppLocalIndex values must not use the values { 1, 3, 5, 6, 7, 8, 9 }.
- AppLocalIndex values { 1, 3, 5, 7 } are automatically assigned and are associated with the identical value of protocolDirLocalIndex. In particular, an entry in the apmAppDirTable indexed by a value provides further information about a protocol indexed by the same value in the protocolDirTable of RMON2.

The value for each supported application must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization, except that if an application is deleted and re-created, it must be re-created with a new value that has not been used since the last re-initialization.

The specific value is meaningful only within a given SNMP entity. An AppLocalIndex value must not be re-used until the next agent restart."

SYNTAX Unsigned32 (1..2147483647)

ProtocolDirNetworkAddress ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A network level address whose semantics and encoding are specified by an associated protocolDirLocalIndex value. Objects of this type must specify which protocolDirLocalIndex value is used. This value is encoded according to the encoding rules for the identified protocolDirectory entry.

For example, if the associated protocolDirLocalIndex indicates an encapsulation of ip, this object is encoded as a length octet of 4, followed by the 4 octets of the ip address, in network byte order.

Objects of this type may allow this value to be the zero length string. If so, they must identify the meaning of this value."

SYNTAX OCTET STRING (SIZE(0..255))

DataSourceOrZero ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Identifies the source of the data that the associated function is configured to analyze. This source can be any interface on this device.

In order to identify a particular interface, this object shall identify the instance of the ifIndex object, defined in [4], for the desired interface.

For example, if an entry were to receive data from interface #1, this object would be set to ifIndex.1.

If the source of the data isn't an interface or cannot be localized to an interface, this object would be set to 0.0"

REFERENCE "The DataSource textual convention is defined in RFC 2021 [5]."

SYNTAX OBJECT IDENTIFIER

RmonClientID ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A long-lived unique ID assigned to an end-system. This ID is assigned by the agent using an implementation-specific algorithm.

Because a client machine may be assigned multiple addresses over any time period it can be difficult to attribute behavior to a particular client based solely on its address. A ClientID may be assigned to provide a more stable handle for referencing that client. The entity that assigns the ClientID may use various implementation techniques to keep track of a client but if the assigning entity is unable to track client address mappings, it may map client identifiers to client addresses rather than to distinct client machines.

This is named ClientID because it helps to solve a problem seen in network clients (servers usually have well-known, long-lived addresses). However, ClientID's may be assigned to any end-system regardless of its role on the network."

SYNTAX Unsigned32 (0..4294967295)

TransactionAggregationType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Specifies one of 4 different techniques for aggregating transactions.

The metrics for a single transaction are the responsiveness of the transaction and whether the transaction succeeded (a boolean). When such metrics are aggregated in this MIB Module, these metrics are replaced by averages and distributions of responsiveness and availability. The metrics describing aggregates are constant no matter which type of aggregation is being performed. These metrics may be found in the `apmReportTable`.

The `flows(1)` aggregation is the simplest. All transactions that share common application/server/client 3-tuples are aggregated together, resulting in a set of metrics for all such unique 3-tuples.

The `clients(2)` aggregation results in somewhat more aggregation (i.e., fewer resulting records). All transactions that share common application/client tuples are aggregated together, resulting in a set of metrics for all such unique tuples.

The `servers(3)` aggregation usually results in still more aggregation (i.e., fewer resulting records). All transactions that share common application/server tuples are aggregated together, resulting in a set of metrics for all such unique tuples.

The `applications(4)` aggregation results in the most aggregation (i.e., the fewest resulting records). All transactions that share a common application are aggregated together, resulting in a set of metrics for all such unique applications.

Note that it is not meaningful to aggregate applications, as different applications have widely varying characteristics. As a result, this set of aggregations is complete."

```
SYNTAX      INTEGER {
                flows(1),      -- Least Aggregation
                clients(2),
                servers(3),
                applications(4) -- Most Aggregation
            }
```

-- The APM Application Directory Group

-- The Application Directory Table contains a record for every

```
-- application monitored by this agent. This table is also used to
-- configure whether or not an application will be measured and which
-- bucket boundaries will be used for the application.
--
-- The bucket boundaries define the break-points between bins of a
-- histogram analysis for that application. As an example of how this
-- works, consider an entry representing response-time for http.
-- If the boundaries are set as follows:
-- Boundary1: 500 milliseconds
-- Boundary2: 1 second
-- Boundary3: 2 seconds
-- Boundary4: 5
-- Boundary5: 15
-- Boundary6: 60
--
-- If the following measurements are made (all in milliseconds):
-- 377, 8645, 1300, 487, 1405, 775, 1115, 850, 945, 1054, 7745, 9380
--
-- A report run during this interval would report the following
-- counts:
-- Bucket1: 2
-- Bucket2: 3
-- Bucket3: 4
-- Bucket4: 0
-- Bucket5: 3
-- Bucket6: 0
-- Bucket7: 0
```

apmAppDirTable OBJECT-TYPE

SYNTAX SEQUENCE OF ApmAppDirEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The APM MIB directory of applications and application verbs. The agent will populate this table with all applications/verbs of any responsivenessType it has the capability to monitor. Since the agent populates this table with every entry it has the capability to monitor, the entries in this table are read-write, allowing the management station to modify parameters in this table but not to add new entries or delete entries (however, entries may be disabled). If new entries are added to the apmHttpFilterTable or the apmUserDefinedAppTable, the agent will add the corresponding entries to this table.

It is an implementation-dependent matter as to how the agent sets these default parameters. For example, it may leave certain entries in this table 'off(0)' if the agent developer

believes that combination will be infrequently used, allowing a manager that needs that capability to set it to 'on(1)'.

Some applications are registered in the RMON2 protocol directory and some are registered in other tables in this MIB Module. Regardless of where an application is originally registered, it is assigned an AppLocalIndex value that is the primary index for this table.

The contents of this table affect all reports and exceptions generated by this agent. Accordingly, modification of this table should be performed by a manager acting in the role of administrator. In particular, management software should not require or enforce particular configuration of this table - it should reflect the preferences of the site administrator, not the software author. As a practical matter, this requires management software to allow the administrator to configure the values it will use so that it can be adapted to the site policy."

```
::= { apmMibObjects 1 }
```

```
apmAppDirEntry OBJECT-TYPE
```

```
SYNTAX      ApmAppDirEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

"The APM MIB directory of applications and application verbs. An entry will exist in this table for all applications for which application performance measurement is supported."

```
INDEX { apmAppDirAppLocalIndex,
        apmAppDirResponsivenessType }
```

```
::= { apmAppDirTable 1 }
```

```
ApmAppDirEntry ::= SEQUENCE {
```

```
    apmAppDirAppLocalIndex      AppLocalIndex,
```

```
    apmAppDirResponsivenessType INTEGER,
```

```
    apmAppDirConfig             INTEGER,
```

```
    apmAppDirResponsivenessBoundary1 Unsigned32,
```

```
    apmAppDirResponsivenessBoundary2 Unsigned32,
```

```
    apmAppDirResponsivenessBoundary3 Unsigned32,
```

```
    apmAppDirResponsivenessBoundary4 Unsigned32,
```

```
    apmAppDirResponsivenessBoundary5 Unsigned32,
```

```
    apmAppDirResponsivenessBoundary6 Unsigned32
```

```
}
```

```
apmAppDirAppLocalIndex OBJECT-TYPE
```

```
SYNTAX      AppLocalIndex
```

```
MAX-ACCESS  not-accessible
```

STATUS current
DESCRIPTION
"The AppLocalIndex assigned for this application Directory entry."
::= { apmAppDirEntry 1 }

apmAppDirResponsivenessType OBJECT-TYPE

SYNTAX INTEGER {
transactionOriented(1),
throughputOriented(2),
streamingOriented(3)
}

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object describes and configures the agent's support for application performance measurement for this application. There are 3 types of measurements for different types of applications:

Transaction-Oriented applications have a fairly constant workload to perform for all transactions. The responsiveness metric for transaction-oriented applications is application response time (from first request to final delivery of service) and is measured in milliseconds. This is commonly referred to as end-user response time.

Throughput-Oriented applications have widely varying workloads based on the nature of the client request. In particular, throughput-oriented applications vary widely in the amount of data that must be transported to satisfy the request. The responsiveness metric for throughput-oriented applications is kilobits per second.

Streaming-Oriented applications deliver data at a constant metered rate of speed regardless of the responsiveness of the networking and computing infrastructure. This constant rate of speed is generally specified to be below (sometimes well below) the nominal capability of the infrastructure. However, when the infrastructures cannot deliver data at this speed, interruption of service or degradation of service can result. The responsiveness metric for streaming-oriented applications is the ratio of time that the service is degraded or interrupted to the total service time. This metric is measured in parts per million.

Note that for some applications, measuring more than one responsiveness type may be interesting. For agents that wish

to support more than one measurement for a application, they will populate this table with multiple entries for that application, one for each type."

::= { apmAppDirEntry 2 }

apmAppDirConfig OBJECT-TYPE

SYNTAX INTEGER {
off(1),
on(2)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object describes and configures support for application performance measurement for this application.

If the value of this object is on(2), the agent supports measurement of application performance metrics for this application and is configured to measure such metrics for all APM MIB functions and all interfaces. If the value of this object is off(1), the agent supports measurement of application performance for this application but is configured to not measure these metrics for any APM MIB functions or interfaces. Whenever this value changes from on(2) to off(1), the agent shall delete all related entries in all tables in this MIB Module.

The value of this object must persist across reboots."

::= { apmAppDirEntry 3 }

apmAppDirResponsivenessBoundary1 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The boundary value between bucket1 and bucket 2. If this value is modified, all entries in the apmReportTable must be deleted by the agent.

The value of this object must persist across reboots."

::= { apmAppDirEntry 4 }

apmAppDirResponsivenessBoundary2 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The boundary value between bucket2 and bucket 3. If this

value is modified, all entries in the apmReportTable must be deleted by the agent.

The value of this object must persist across reboots."
 ::= { apmAppDirEntry 5 }

apmAppDirResponsivenessBoundary3 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The boundary value between bucket3 and bucket 4. If this value is modified, all entries in the apmReportTable must be deleted by the agent.

The value of this object must persist across reboots."
 ::= { apmAppDirEntry 6 }

apmAppDirResponsivenessBoundary4 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The boundary value between bucket4 and bucket 5. If this value is modified, all entries in the apmReportTable must be deleted by the agent.

The value of this object must persist across reboots."
 ::= { apmAppDirEntry 7 }

apmAppDirResponsivenessBoundary5 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The boundary value between bucket5 and bucket 6. If this value is modified, all entries in the apmReportTable must be deleted by the agent.

The value of this object must persist across reboots."
 ::= { apmAppDirEntry 8 }

apmAppDirResponsivenessBoundary6 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The boundary value between bucket6 and bucket 7. If this

value is modified, all entries in the apmReportTable must be deleted by the agent.

The value of this object must persist across reboots."
 ::= { apmAppDirEntry 9 }

-- Scalars related to the Application Directory table

apmBucketBoundaryLastChange OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime the last time that any bucket boundary in any appDirEntry was changed. This object can help to determine if two managers are both trying to enforce different configurations of this table."

::= { apmMibObjects 2 }

apmAppDirID OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object allows managers to avoid downloading application directory information when the directory is set to a known (usually fixed) configuration.

If the value of this object isn't 0.0, it signifies that the entire contents of the apmAppDirTable, apmHttpFilterTable, apmUserDefinedAppTable and protocolDirTable are equal to a known state identified by the value of this object. If a manager recognizes this value as identifying a directory configuration it has a local copy of, it may use this local copy rather than downloading these tables. Note that it may have downloaded this local copy (and the ID) from another agent and used this copy for all other agents that advertised the same ID.

If an agent recognizes that the entire contents of the apmAppDirTable, apmHttpFilterTable, apmUserDefinedAppTable and protocolDirTable are equal to a known state to which an ID has been assigned, it should set this object to that ID.

In many cases when this feature is used, the application directory information will be in read-only memory and thus the tables may not be modified via SNMP requests. In the event

that the tables are writable and a modification is made, the agent is responsible for setting this object to 0.0 if it cannot determine that the state is equal to a known state.

An agent is not obligated to recognize and advertise all such registered states as it may not have knowledge of all states. Thus, a manager may encounter agents whose DirectoryID value is 0.0 even though the contents of the directory were equal to a registered state.

Note that the contents of those tables includes the protocolDirLocalIndex and appLocalIndex values. In other words, these values can't be assigned randomly on each agent, but must be equal to values that are part of the known state. While it is possible for a manager to download application directory details using SNMP and to set the appropriate directoryID, the manager would need to have some scheme to ensure consistent values of LocalIndex variables from agent to agent. Such schemes are outside the scope of this specification.

Application directory registrations are unique within an administrative domain.

Typically these registrations will be made by an agent software developer who will set the application directory tables to a read-only state and assign a DirectoryID to that state. Thus, all agents running this software would share the same DirectoryID. As the application directory might change from one software release to the next, the developer may register different DirectoryID's for each software release.

A customer could also create a site-wide application directory configuration and assign a DirectoryID to that configuration as long as consistent values of LocalIndex variables can be ensured.

The value of this object must persist across reboots."
 ::= { apmMibObjects 3 }

-- APM HTTP Filter Table

-- The HTTP Filter Table creates virtual applications which measure the
 -- performance of certain web pages or sets of web pages. Some
 -- circumstances where this is particularly useful are:

--
 -- - An Intranet or ASP scenario where a business application is
 -- running on one or more web pages or scripts.

```
--      (i.e., /expense/submit.cgi?employeeID=3426&...)
--      - A web-hosting scenario where one wants to measure the
--        service level for a particular customer
--      - An e-commerce scenario where the performance of certain
--        pages needs to be monitored more closely.
--      (i.e., shopping cart, shipping, credit card authorization)
```

apmHttpFilterTable OBJECT-TYPE

```
SYNTAX      SEQUENCE OF ApmHttpFilterEntry
MAX-ACCESS  not-accessible
STATUS      current
```

DESCRIPTION

"A table that creates virtual applications which measure the performance of certain web pages or sets of web pages.

When an entry is added to this table, the agent will automatically create one or more entries in the apmAppDirTable (one for each responsivenessType it is capable of measuring).

Note that when entries exist in this table some HTTP transactions will be summarized twice: in applications represented here as well as the HTTP application. If entries in this table overlap, these transactions may be summarized additional times.

The contents of this table affect all reports and exceptions generated by this agent. Accordingly, modification of this table should be performed by a manager acting in the role of administrator. In particular, management software should not require or enforce particular configuration of this table - it should reflect the preferences of the site administrator, not the software author."

```
::= { apmMibObjects 4 }
```

apmHttpFilterEntry OBJECT-TYPE

```
SYNTAX      ApmHttpFilterEntry
MAX-ACCESS  not-accessible
STATUS      current
```

DESCRIPTION

"A virtual application which measure the performance of certain web pages or sets of web pages."

```
INDEX { apmHttpFilterIndex }
```

```
::= { apmHttpFilterTable 1 }
```

```
ApmHttpFilterEntry ::= SEQUENCE {
```

```
    apmHttpFilterIndex      Unsigned32,
    apmHttpFilterAppLocalIndex AppLocalIndex,
```

```

    apmHttpFilterServerProtocol      Unsigned32,
    apmHttpFilterServerAddress      ProtocolDirNetworkAddress,
    apmHttpFilterURLPath            OCTET STRING,
    apmHttpFilterMatchType          INTEGER,
    apmHttpFilterOwner              OwnerString,
    apmHttpFilterStorageType        StorageType,
    apmHttpFilterRowStatus          RowStatus
}

```

apmHttpFilterIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An index that uniquely identifies an entry in the
apmHttpFilterTable."

::= { apmHttpFilterEntry 1 }

apmHttpFilterAppLocalIndex OBJECT-TYPE

SYNTAX AppLocalIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The AppLocalIndex that represents HTTP transactions
that match this entry."

This object is read-only. A value is created by the agent from
an unused AppLocalIndex value when this apmHttpFilterEntry is
created."

::= { apmHttpFilterEntry 2 }

apmHttpFilterServerProtocol OBJECT-TYPE

SYNTAX Unsigned32 (1..2147483647)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The protocolDirLocalIndex value of the network level protocol
of the apmHttpFilterServerAddress."

::= { apmHttpFilterEntry 3 }

apmHttpFilterServerAddress OBJECT-TYPE

SYNTAX ProtocolDirNetworkAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This entry will only represent transactions coming from the
network address specified in this object."

This is represented as an octet string with specific semantics and length as identified by the associated `apmHttpFilterServerProtocol` object.

If this object is the zero-length string, then this entry will match one of the addresses represented by the 'host' component of the associated `apmHttpFilterURLPath` object, where the format if a URL [9] is
`http://<host>:<port>/<path>?<searchpart>."`

`::= { apmHttpFilterEntry 4 }`

`apmHttpFilterURLPath` OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..65535))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This entry will only represent HTTP transactions where the URL path component in the request matches this value. This value represents the requested path regardless of any substitution that the server might perform.

Prior to the matching, the URL is stripped of any server address or DNS name and consists solely of the path name on that server.

If the length of this object is zero, then this entry will match if the associated `apmHttpFilterServerAddress` match. If the length of that object is also zero, then this entry will match nothing.

The value of the associated `apmHttpFilterMatchType` dictates the type of matching that will be attempted."

`::= { apmHttpFilterEntry 5 }`

`apmHttpFilterMatchType` OBJECT-TYPE

SYNTAX INTEGER {
 exact(1),
 stripTrailingSlash(2),
 prefix(3)
 }

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The matching algorithm used to compare the URL pathname.

If the value is `exact(1)`, then the pathname component will be compared with the associated `apmHttpFilterURLPath` and will only be associated with this entry if it matches exactly.

If the value is stripTrailingSlash(2), then the pathname component will be compared with the associated apmHttpFilterURLPath and will only be associated with this entry if it matches exactly or if the pathname ends with a '/' symbol and matches apmHttpFilterURLPath if the '/' symbol is removed from the pathname. This option exists for those paths where an optional trailing slash is possible but for which a prefix match would be too broad.

If the value is prefix(3), then the pathname component will be compared with the associated apmHttpFilterURLPath and will only be associated with this entry if the beginning of the pathname matches every octet of this value. Octets that extend beyond the length of this value are ignored."

::= { apmHttpFilterEntry 6 }

apmHttpFilterOwner OBJECT-TYPE

SYNTAX OwnerString
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The entity that configured this entry and is therefore using the resources assigned to it."

::= { apmHttpFilterEntry 7 }

apmHttpFilterStorageType OBJECT-TYPE

SYNTAX StorageType
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The storage type of this apmHttpFilterEntry. If the value of this object is 'permanent', no objects in this row need to be writable."

::= { apmHttpFilterEntry 8 }

apmHttpFilterRowStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The status of this apmHttpFilterEntry. No objects in this row may be modified while the row's status is 'active'."

::= { apmHttpFilterEntry 9 }

apmHttpIgnoreUnregisteredURLs OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current

DESCRIPTION

"When true, APM measurements of HTTP transactions will only measure transactions relating to URLs that match a filter in the apmHttpFilterTable. Thus, measurements for the HTTP application will present aggregated statistics for URL-matching HTTP transactions and measurements for the HTTP GET application verb will present aggregated statistics for URL-matching HTTP GET transactions.

This will be used in environments that wish to monitor only targeted URLs and to ignore large volumes of internet web browsing traffic.

This object affects all APM reports and exceptions generated by this agent. Accordingly, modification of this object should be performed by a manager acting in the role of administrator. In particular, management software should not require or enforce particular configuration of this object - it should reflect the preferences of the site administrator, not the software author.

The value of this object must persist across reboots."

::= { apmMibObjects 5 }

apmHttp4xxIsFailure OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"When true, this agent will recognize HTTP errors in the range of 400 through 499 and will treat them as unavailable transactions. When false or when this object isn't supported, they will be treated as successful transactions.

This object allows such error pages to be tracked at the possible expense of having user typos treated as poor service on the part of the web server.

This object affects all reports and exceptions generated by this agent. Accordingly, modification of this object should be performed by a manager acting in the role of administrator. In particular, management software should not require or enforce particular configuration of this object - it should reflect the preferences of the site administrator, not the software author.

The value of this object must persist across reboots."

::= { apmMibObjects 6 }

-- The APM User-Defined Application Table

-- Many application protocols will never be registered with a
 -- standards body (and thus included in a protocol directory standard)
 -- because they are custom, in-house or proprietary
 -- applications. Nevertheless, implementation strategies exist for
 -- monitoring the end-user experience of these applications.
 --
 -- This read-only table provides a means for the agent to advertise
 -- which user-defined applications it is monitoring and to associate
 -- each with an AppLocalIndex value. It is an implementation-dependent
 -- matter as to how the agent learns how to monitor these
 -- applications.

apmUserDefinedAppTable OBJECT-TYPE

SYNTAX SEQUENCE OF ApmUserDefinedAppEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table that advertises user-defined applications that the agent is measuring.

The agent will automatically create one or more entries in the apmAppDirTable (one for each responsivenessType it is capable of measuring) for each entry in this table.

Note that when entries exist in this table some transactions can be summarized more than once if there is overlap between applications defined here and applications defined in the protocol directory or in the httpFilter table."

::= { apmMibObjects 7 }

apmUserDefinedAppEntry OBJECT-TYPE

SYNTAX ApmUserDefinedAppEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A user-defined application that the agent is measuring, along with its AppLocalIndex assignment.

The apmAppDirAppLocalIndex value in the index identifies the agent-assigned AppLocalIndex value for this user-defined application."

INDEX { apmAppDirAppLocalIndex }

::= { apmUserDefinedAppTable 1 }

ApmUserDefinedAppEntry ::= SEQUENCE {
 apmUserDefinedAppParentIndex Unsigned32,

```

    apmUserDefinedAppApplication      SnmpAdminString
}

```

```

apmUserDefinedAppParentIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The protocolDirLocalIndex value of the highest-layer
        protocol defined in the protocolDirTable that this
        application is a child of."
    ::= { apmUserDefinedAppEntry 1 }

```

```

apmUserDefinedAppApplication OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A human readable descriptive tag for this application."
    ::= { apmUserDefinedAppEntry 2 }

```

-- The APM Name Table

```

apmNameTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ApmNameEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A client machine may have multiple addresses during a period
        of monitoring. The apmNameTable assigns a long-lived
        identifier to a client and records what addresses were
        assigned to that client for periods of time. Various
        implementation techniques exist for tracking this mapping but
        if an agent is unable to track client address mappings, it may
        map client identifiers to client addresses rather than to
        distinct client machines.

```

A particular apmNameClientID should be a constant attribute of a particular client. When available, the agent may also record the machine name and/or user name which may be valuable for displaying to humans. The apmNameMachineName and apmNameUserName are relatively constant, changing only if these attributes actually change on the client.

The agent will store a historical log of these entries, aging out old entries as the log becomes too large. Since this table contains information vital to the interpretation of other tables (e.g., the apmReportTable), the agent should ensure that

the log doesn't age out entries that would be referenced by data in those tables.

Note that an entry for a clientID is active from its StartTime until the StartTime of another entry (for the same clientID) that supersedes it, or 'now' if none supersedes it. Therefore, if a clientID only has a single entry, it is by definition very new and should never be aged out. No entry for a clientID should be aged out unless it has been updated by a new entry for the client (i.e., with an updated address) and only if the new entry is 'old' enough.

To determine how old is old enough, compute the maximum value of Interval * (NumReports + 1) of all entries in the apmReportControlTable (the '+ 1' is to allow a reasonable period of time for the report to be downloaded). Then take the larger of this value and the age in seconds of the oldest entry in the current transaction table. If an entry for a clientID is superseded by another entry whose StartTime is more than this many seconds ago, then the older entry may be deleted."

```
::= { apmMibObjects 8 }
```

apmNameEntry OBJECT-TYPE

```
SYNTAX      ApmNameEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

"An entry in the APM name table. An entry exists for each period of time that a client has been associated with a particular address.

The protocolDirLocalIndex value in the index identifies the network layer protocol for the ClientAddress for this entry.

Note that some combinations of index values may result in an index that exceeds 128 sub-identifiers in length which exceeds the maximum for the SNMP protocol. Implementations should take care to avoid such combinations."

```
INDEX { apmNameClientID,
        protocolDirLocalIndex, apmNameClientAddress,
        apmNameMappingStartTime }
::= { apmNameTable 1 }
```

```
ApmNameEntry ::= SEQUENCE {
    apmNameClientID          RmonClientID,
    apmNameClientAddress     ProtocolDirNetworkAddress,
```

```

    apmNameMappingStartTime      DateAndTime,
    apmNameMachineName          SnmpAdminString,
    apmNameUserName              SnmpAdminString
}

```

```

apmNameClientID OBJECT-TYPE
    SYNTAX      RmonClientID
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A unique ID assigned to the machine represented by this
        mapping. This ID is assigned by the agent using an
        implementation-specific algorithm."
    ::= { apmNameEntry 1 }

```

```

apmNameClientAddress OBJECT-TYPE
    SYNTAX      ProtocolDirNetworkAddress (SIZE(1..255))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The network client address for this client when this mapping
        was active.

        This is represented as an octet string with specific semantics
        and length as identified by the protocolDirLocalIndex
        component of the index. This object may not be the zero length
        string.

        Since this object is an index variable, it is encoded in the
        index according to the index encoding rules. For example, if
        the protocolDirLocalIndex component of the index indicates an
        encapsulation of ip, this object is encoded as a length octet
        of 4, followed by the 4 octets of the ip address, in network
        byte order. Care should be taken to avoid values of this
        object that, in conjunction with the other index variables,
        would result in an index longer than SNMP's maximum of 128
        subidentifiers."
    ::= { apmNameEntry 2 }

```

```

apmNameMappingStartTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The time that the agent first discovered this mapping
        as active."
    ::= { apmNameEntry 3 }

```

apmNameMachineName OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The human readable name of the client machine.

If the client has no machine name or the agent is unable to learn the machine name, this object will be a zero-length string."

::= { apmNameEntry 4 }

apmNameUserName OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The human readable name of a human user using the client machine. If more than one user name are available simultaneously, it is an implementation-dependent matter as to which is used here. However, if the user name changes, this object should change to reflect that change.

Non-human user names like 'root' or 'administrator' aren't intended as values for this object. If the client has no recorded user name or the agent is unable to learn a user name, this object will be a zero-length string."

::= { apmNameEntry 5 }

-- The APM Report Group**apmReportControlTable OBJECT-TYPE**

SYNTAX SEQUENCE OF ApmReportControlEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Parameters that control the creation of a set of reports that aggregate application performance."

::= { apmMibObjects 9 }

apmReportControlEntry OBJECT-TYPE

SYNTAX ApmReportControlEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual row in the apmReportControlTable.

An example of the indexing of this table is

```

    apmReportControlInterval.3"
INDEX { apmReportControlIndex }
 ::= { apmReportControlTable 1 }

```

```

ApmReportControlEntry ::= SEQUENCE {
    apmReportControlIndex          Unsigned32,
    apmReportControlDataSource     DataSourceOrZero,
    apmReportControlAggregationType TransactionAggregationType,
    apmReportControlInterval      Unsigned32,
    apmReportControlRequestedSize  Unsigned32,
    apmReportControlGrantedSize   Unsigned32,
    apmReportControlRequestedReports Unsigned32,
    apmReportControlGrantedReports Unsigned32,
    apmReportControlStartTime     TimeStamp,
    apmReportControlReportNumber  Unsigned32,
    apmReportControlDeniedInserts Counter32,
    apmReportControlDroppedFrames Counter32,
    apmReportControlOwner         OwnerString,
    apmReportControlStorageType    StorageType,
    apmReportControlStatus         RowStatus
}

```

```

apmReportControlIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index that uniquely identifies an entry in the
        apmReportControlTable. Each such entry defines a unique
        report whose results are placed in the apmReportTable on
        behalf of this apmReportControlEntry."
    ::= { apmReportControlEntry 1 }

```

```

apmReportControlDataSource OBJECT-TYPE
    SYNTAX      DataSourceOrZero
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The source of the data for APM Reports generated on
        behalf of this apmReportControlEntry.

```

If the measurement is being performed by a probe, this should be set to interface or port where data was received for analysis. If the measurement isn't being performed by a probe, this should be set to the primary interface over which the measurement is being performed. If the measurement isn't being performed by a probe and there is no primary interface or this

information isn't known, this object should be set to 0.0.

This object may not be modified if the associated
apmReportControlStatus object is equal to active(1)."
::= { apmReportControlEntry 2 }

apmReportControlAggregationType OBJECT-TYPE

SYNTAX TransactionAggregationType

```
--    INTEGER {
--        flows(1),
--        clients(2),
--        servers(3),
--        applications(4)
--    }
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The type of aggregation being performed for this set of reports.

The metrics for a single transaction are the responsiveness of the transaction and whether the transaction succeeded (a boolean). When such metrics are aggregated in this MIB Module, these metrics are replaced by averages and distributions of responsiveness and availability. The metrics describing aggregates are constant no matter which type of aggregation is being performed. These metrics may be found in the apmReportTable.

The flows(1) aggregation is the simplest. All transactions that share common application/server/client 3-tuples are aggregated together, resulting in a set of metrics for all such unique 3-tuples.

The clients(2) aggregation results in somewhat more aggregation (i.e., fewer resulting records). All transactions that share common application/client tuples are aggregated together, resulting in a set of metrics for all such unique tuples.

The servers(3) aggregation usually results in still more aggregation (i.e., fewer resulting records). All transactions that share common application/server tuples are aggregated together, resulting in a set of metrics for all such unique tuples.

The applications(4) aggregation results in the most aggregation (i.e., the fewest resulting records). All

transactions that share a common application are aggregated together, resulting in a set of metrics for all such unique applications.

Note that it is not meaningful to aggregate applications, as different applications have widely varying characteristics. As a result, this set of aggregations is complete.

This object may not be modified if the associated
apmReportControlStatus object is equal to active(1)."
::= { apmReportControlEntry 3 }

apmReportControlInterval OBJECT-TYPE

SYNTAX Unsigned32

UNITS "Seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The interval in seconds over which data is accumulated before being aggregated into a report in the apmReportTable. All reports with the same apmReportControlIndex will be based on the same interval. This object must be greater than zero.

Many users desire that these reports be synchronized to within seconds of the beginning of the hour because the results may be correlated more meaningfully to business behavior and so that data from multiple agents is aggregated over the same time periods. Thus management software may take extra effort to synchronize reports to the beginning of the hour and to one another. However, the agent must not allow reports to 'drift' over time as they will quickly become unsynchronized. In particular, if there is any fixed processing delay between reports, the reports should deduct this time from the interval so that reports don't drift.

This object may not be modified if the associated
apmReportControlStatus object is equal to active(1)."
DEFVAL { 3600 }
::= { apmReportControlEntry 4 }

apmReportControlRequestedSize OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The number of entries requested to be allocated for each report generated on behalf of this entry."

::= { apmReportControlEntry 5 }

apmReportControlGrantedSize OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of entries per report the agent has allocated based on the requested amount in apmReportControlRequestedSize. Since multiple reports are saved, the total number of entries allocated will be this number multiplied by the value of apmReportControlGrantedReports, or 1 if that object doesn't exist.

When the associated apmReportControlRequestedSize object is created or modified, the agent should set this object as closely to the requested value as is possible for the particular implementation and available resources. When considering resources available, the agent must consider its ability to allocate this many entries for all reports.

Note that while the actual number of entries stored in the reports may fluctuate due to changing conditions, the agent must continue to have storage available to satisfy the full report size for all reports when necessary. Further, the agent must not lower this value except as a result of a set to the associated apmReportControlRequestedSize object."

::= { apmReportControlEntry 6 }

apmReportControlRequestedReports OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The number of saved reports requested to be allocated on behalf of this entry."

::= { apmReportControlEntry 7 }

apmReportControlGrantedReports OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of saved reports the agent has allocated based on the requested amount in apmReportControlRequestedReports. Since each report can have many entries, the total number of entries allocated will be this number multiplied by the value of apmReportControlGrantedSize, or 1 if that object doesn't exist.

When the associated `apmReportControlRequestedReports` object is created or modified, the agent should set this object as closely to the requested value as is possible for the particular implementation and available resources. When considering resources available, the agent must consider its ability to allocate this many reports each with the number of entries represented by `apmReportControlGrantedSize`, or 1 if that object doesn't exist.

Note that while the storage required for each report may fluctuate due to changing conditions, the agent must continue to have storage available to satisfy the full report size for all reports when necessary. Further, the agent must not lower this value except as a result of a set to the associated `apmReportControlRequestedSize` object."

::= { apmReportControlEntry 8 }

`apmReportControlStartTime` OBJECT-TYPE

SYNTAX Timestamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of `sysUpTime` when the system began processing the report in progress. Note that the report in progress is not available.

This object may be used by the management station to figure out the start time for all previous reports saved for this `apmReportControlEntry`, as reports are started at fixed intervals."

::= { apmReportControlEntry 9 }

`apmReportControlReportNumber` OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of the report in progress. When an `apmReportControlEntry` is activated, the first report will be numbered one."

::= { apmReportControlEntry 10 }

`apmReportControlDeniedInserts` OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of failed attempts to add an entry to reports for

this apmReportControlEntry because the number of entries would have exceeded apmReportControlGrantedSize.

This number is valuable in determining if enough entries have been allocated for reports in light of fluctuating network usage. Note that since an entry that is denied will often be attempted again, this number will not predict the exact number of additional entries needed, but can be used to understand the relative magnitude of the problem.

Also note that there is no ordering specified for the entries in the report, thus there are no rules for which entries will be omitted when not enough entries are available. As a consequence, the agent is not required to delete 'least valuable' entries first."

```
::= { apmReportControlEntry 11 }
```

apmReportControlDroppedFrames OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of frames which were received by the agent and therefore not accounted for in the *StatsDropEvents, but for which the agent chose not to count for this entry for whatever reason. Most often, this event occurs when the agent is out of some resources and decides to shed load from this collection.

This count does not include packets that were not counted because they had MAC-layer errors.

This counter is only relevant if this apm report is based on a data source whose collection methodology is based on analyzing network traffic.

Note that if the apmReportTables are inactive because no applications are enabled in the application directory, this value should be 0.

Note that, unlike the dropEvents counter, this number is the exact number of frames dropped."

```
::= { apmReportControlEntry 12 }
```

apmReportControlOwner OBJECT-TYPE

SYNTAX OwnerString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The entity that configured this entry and is therefore using the resources assigned to it."

::= { apmReportControlEntry 13 }

apmReportControlStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type of this apmReportControlEntry. If the value of this object is 'permanent', no objects in this row need to be writable."

::= { apmReportControlEntry 14 }

apmReportControlStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this apmReportControlEntry.

An entry may not exist in the active state unless all objects in the entry have an appropriate value. The only objects in the entry that may be modified while the entry is in the active state are apmReportControlRequestedSize and apmReportControlRequestedReports.

If this object is not equal to active(1), all associated entries in the apmReportTable shall be deleted by the agent."

::= { apmReportControlEntry 15 }

-- The APM Report Table

apmReportTable OBJECT-TYPE

SYNTAX SEQUENCE OF ApmReportEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The data resulting from aggregated APM reports. Consult the definition of apmReportControlAggregationType for the definition of the various types of aggregations."

::= { apmMibObjects 10 }

apmReportEntry OBJECT-TYPE

SYNTAX ApmReportEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual row in the apmReportTable.

The apmReportControlIndex value in the index identifies the apmReportControlEntry on whose behalf this entry was created. The apmReportIndex value in the index identifies which report (in the series of reports) this entry is a part of.

The apmAppDirAppLocalIndex value in the index identifies the common application of the transactions aggregated in this entry.

The apmAppDirResponsivenessType value in the index identifies the type of responsiveness metric reported by this entry and uniquely identifies this entry when more than one responsiveness metric is measured for a flow. Entries will only exist in this table for those combinations of AppLocalIndex and ResponsivenessType that are configured 'on(1)'.

The protocolDirLocalIndex value in the index identifies the network layer protocol of the apmReportServerAddress. When the associated apmReportControlAggregationType value is equal to applications(4) or clients(2), this protocolDirLocalIndex value will equal 0.

The apmReportServerAddress value in the index identifies the network layer address of the server in transactions aggregated in this entry.

The apmNameClientID value in the index identifies the client in transactions aggregated in this entry. If the associated apmReportControlAggregationType is equal to applications(4) or servers(3), then this protocolDirLocalIndex value will equal 0.

An example of the indexing of this entry is

apmReportTransactionCount.3.15.3.1.8.4.192.168.1.2.3232235788

Note that some combinations of index values may result in an index that exceeds 128 sub-identifiers in length which exceeds the maximum for the SNMP protocol. Implementations should take care to avoid such combinations."

```
INDEX { apmReportControlIndex, apmReportIndex,
        apmAppDirAppLocalIndex,
        apmAppDirResponsivenessType,
        protocolDirLocalIndex, apmReportServerAddress,
        apmNameClientID }
 ::= { apmReportTable 1 }
```

```
ApmReportEntry ::= SEQUENCE {
    apmReportIndex                Unsigned32,
    apmReportServerAddress        ProtocolDirNetworkAddress,
```

```

    apmReportTransactionCount      Unsigned32,
    apmReportSuccessfulTransactions Unsigned32,
    apmReportResponsivenessMean    Unsigned32,
    apmReportResponsivenessMin     Unsigned32,
    apmReportResponsivenessMax     Unsigned32,
    apmReportResponsivenessB1      Unsigned32,
    apmReportResponsivenessB2      Unsigned32,
    apmReportResponsivenessB3      Unsigned32,
    apmReportResponsivenessB4      Unsigned32,
    apmReportResponsivenessB5      Unsigned32,
    apmReportResponsivenessB6      Unsigned32,
    apmReportResponsivenessB7      Unsigned32
}

```

apmReportIndex OBJECT-TYPE

```

SYNTAX      Unsigned32 (1..4294967295)
MAX-ACCESS  not-accessible
STATUS      current

```

DESCRIPTION

"The value of apmReportControlReportNumber for the report to which this entry belongs."

```
 ::= { apmReportEntry 1 }
```

apmReportServerAddress OBJECT-TYPE

```

SYNTAX      ProtocolDirNetworkAddress
MAX-ACCESS  not-accessible
STATUS      current

```

DESCRIPTION

"The network server address for this apmReportEntry."

This is represented as an octet string with specific semantics and length as identified by the protocolDirLocalIndex component of the index.

Since this object is an index variable, it is encoded in the index according to the index encoding rules. For example, if the protocolDirLocalIndex indicates an encapsulation of ip, this object is encoded as a length octet of 4, followed by the 4 octets of the ip address, in network byte order. Care should be taken to avoid values of this object that, in conjunction with the other index variables, would result in an index longer than SNMP's maximum of 128 subidentifiers.

If the associated apmReportControlAggregationType is equal to applications(4) or clients(2), then this object will be a null string and will be encoded simply as a length octet of 0."

```
 ::= { apmReportEntry 2 }
```

apmReportTransactionCount OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of transactions aggregated into this record."

::= { apmReportEntry 3 }

apmReportSuccessfulTransactions OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of successful transactions aggregated into this record."

::= { apmReportEntry 4 }

apmReportResponsivenessMean OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The arithmetic mean of the responsiveness metrics for all successful transactions aggregated into this record."

::= { apmReportEntry 5 }

apmReportResponsivenessMin OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The minimum of the responsiveness metrics for all successful transactions aggregated into this record."

::= { apmReportEntry 6 }

apmReportResponsivenessMax OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum of the responsiveness metrics for all successful transactions aggregated into this record."

::= { apmReportEntry 7 }

-- Note that when updating a report entry, a transaction will not be
-- counted in more than 1 bucket in an entry. It will be counted in
-- the first bucket that matches, starting with Bucket 1 (B1). Note
-- that if a transaction matches 2 application types, it will update

-- one bucket in each of 2 entries in this table.

apmReportResponsivenessB1 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of successful transactions aggregated into this record whose responsiveness was less than boundary1 value for this application."

::= { apmReportEntry 8 }

apmReportResponsivenessB2 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of successful transactions aggregated into this record whose responsiveness did not fall into Bucket 1 and was greater than or equal to the boundary1 value for this application and less than the boundary2 value for this application."

::= { apmReportEntry 9 }

apmReportResponsivenessB3 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of successful transactions aggregated into this record whose responsiveness did not fall into Bucket 1 or 2 and as greater than or equal to the boundary2 value for this application and less than the boundary3 value for this application."

::= { apmReportEntry 10 }

apmReportResponsivenessB4 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of successful transactions aggregated into this record whose responsiveness did not fall into Buckets 1 through 3 and was greater than or equal to the boundary3 value for this application and less than the boundary4 value for this application."

::= { apmReportEntry 11 }

apmReportResponsivenessB5 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of successful transactions aggregated into this record whose responsiveness did not fall into Buckets 1 through 4 and was greater than or equal to the boundary4 value for this application and less than the boundary5 value for this application."

::= { apmReportEntry 12 }

apmReportResponsivenessB6 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of successful transactions aggregated into this record whose responsiveness did not fall into Buckets 1 through 5 and was greater than or equal to the boundary5 value for this application and less than the boundary6 value for this application."

::= { apmReportEntry 13 }

apmReportResponsivenessB7 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of successful transactions aggregated into this record whose responsiveness did not fall into Buckets 1 through 6 and was greater than or equal to the boundary6 value for this application."

::= { apmReportEntry 14 }

-- APM Transaction Table

apmTransactionTable OBJECT-TYPE

SYNTAX SEQUENCE OF ApmTransactionEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains transactions that are currently running or have recently finished."

::= { apmMibObjects 11 }

apmTransactionEntry OBJECT-TYPE

SYNTAX ApmTransactionEntry

MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"A conceptual row in the apmTransactionTable.

The apmAppDirAppLocalIndex value in the index identifies the application of the transaction represented by this entry. The apmAppDirResponsivenessType value in the index identifies the type of responsiveness metric reported by this entry and uniquely identifies this entry when more than one responsiveness metric is measured for a flow. Entries will only exist in this table for those combinations of AppLocalIndex and ResponsivenessType that are configured 'on(1)'. The protocolDirLocalIndex value in the index identifies the network layer protocol of the apmTransactionServerAddress. The apmTransactionServerAddress value in the index identifies the network layer address of the server in the transaction represented by this entry. The apmNameClientID value in the index identifies the client in the transaction represented by this entry.

An example of the indexing of this entry is
 apmTransactionCount.3.1.8.4.192.168.1.2.3232235788.2987

Note that some combinations of index values may result in an index that exceeds 128 sub-identifiers in length which exceeds the maximum for the SNMP protocol. Implementations should take care to avoid such combinations."

```
INDEX { apmAppDirAppLocalIndex,
        apmAppDirResponsivenessType,
        protocolDirLocalIndex, apmTransactionServerAddress,
        apmNameClientID, apmTransactionID }
 ::= { apmTransactionTable 1 }
```

```
ApmTransactionEntry ::= SEQUENCE {
    apmTransactionServerAddress    ProtocolDirNetworkAddress,
    apmTransactionID              Unsigned32,
    apmTransactionResponsiveness  Unsigned32,
    apmTransactionAge             TimeInterval,
    apmTransactionSuccess         TruthValue
}
```

```
apmTransactionServerAddress OBJECT-TYPE
    SYNTAX      ProtocolDirNetworkAddress (SIZE (1..255))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

"The network server address for this apmTransactionEntry.

This is represented as an octet string with specific semantics and length as identified by the protocolDirLocalIndex component of the index. This object may not be the zero length string.

For example, if the protocolDirLocalIndex indicates an encapsulation of ip, this object is encoded as a length octet of 4, followed by the 4 octets of the ip address, in network byte order. Care should be taken to avoid values of this object that, in conjunction with the other index variables, would result in an index longer than SNMP's maximum of 128 subidentifiers."

::= { apmTransactionEntry 1 }

apmTransactionID OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A unique value for this transaction amongst other transactions sharing the same application layer protocol and server and client addresses. Implementations may choose to use the value of the client's source port, when possible."

::= { apmTransactionEntry 2 }

apmTransactionResponsiveness OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The current value of the responsiveness metric for this transaction. If this transaction has completed, the final value of the metric will be available.

Note that this value may change over the lifetime of the transaction and it is the final value of this metric that is recorded as the responsiveness of the transaction for use in other APM MIB functions."

::= { apmTransactionEntry 3 }

apmTransactionAge OBJECT-TYPE

SYNTAX TimeInterval

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If this transaction is still executing, this value shall be

the length of time since it was started. If it has completed, this value shall be the length of time it was executing."
 ::= { apmTransactionEntry 4 }

apmTransactionSuccess OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The success of this transaction up to this time. Once a transaction has been marked as failed, it cannot move back into the successful state."

::= { apmTransactionEntry 5 }

apmTransactionsRequestedHistorySize OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The maximum number of completed transactions desired to be retained in the apmTransactionTable. If the agent doesn't have enough resources to retain this many, it will retain as many as possible. Regardless of this value, the agent must attempt to keep records for all current transactions it is monitoring.

The value of this object must persist across reboots."

::= { apmMibObjects 12 }

```
-- The APM Exception table
-- The APM Exception Table creates filters so that a management
-- station can get immediate notification of a transaction that has
-- had poor availability or responsiveness.
--
-- This function is particularly helpful in unaggregated situations
-- where the numbers of agents is relatively high and the transaction
-- rate per agent is relatively low (such as agents for desktops or
-- dedicated to small workgroups). Polling agents in such an
-- environment would either cause scalability problems (high rate) or
-- lead to long notification delays (low rate).
```

apmExceptionTable OBJECT-TYPE

SYNTAX SEQUENCE OF ApmExceptionEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table creates filters so that a management station can get immediate notification of a transaction that has had poor

availability or responsiveness.

Each `apmExceptionEntry` is associated with a particular type of transaction and is applied to all transactions of that type. Multiple `apmExceptionEntries` may be associated with a particular type of transaction. A transaction type is identified by the value of the `apmAppDirAppLocalIndex` component of the index.

Because the quality of a transaction is not known until it is completed, these thresholds are only applied after the transaction has completed."

```
::= { apmMibObjects 13 }
```

`apmExceptionEntry` OBJECT-TYPE

SYNTAX `ApmExceptionEntry`

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual row in the `apmExceptionTable`.

The `apmAppDirAppLocalIndex` value in the index identifies the application this entry will monitor.

The `apmAppDirResponsivenessType` value in the index identifies the type of responsiveness metric this entry will monitor."

```
INDEX { apmAppDirAppLocalIndex,
        apmAppDirResponsivenessType, apmExceptionIndex }
::= { apmExceptionTable 1 }
```

`ApmExceptionEntry` ::= SEQUENCE {

<code>apmExceptionIndex</code>	Unsigned32,
<code>apmExceptionResponsivenessComparison</code>	INTEGER,
<code>apmExceptionResponsivenessThreshold</code>	Unsigned32,
<code>apmExceptionUnsuccessfulException</code>	INTEGER,
<code>apmExceptionResponsivenessEvents</code>	Counter32,
<code>apmExceptionUnsuccessfulEvents</code>	Counter32,
<code>apmExceptionOwner</code>	OwnerString,
<code>apmExceptionStorageType</code>	StorageType,
<code>apmExceptionStatus</code>	RowStatus

}

`apmExceptionIndex` OBJECT-TYPE

SYNTAX Unsigned32 (1..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An index that uniquely identifies an entry in the apmExceptionTable amongst other entries with equivalent index values for apmAppDirAppLocalIndex and apmAppDirResponsivenessType. Each such entry sets up thresholds for a particular measurement of a particular application."

::= { apmExceptionEntry 1 }

apmExceptionResponsivenessComparison OBJECT-TYPE

SYNTAX INTEGER {
none(1),
greater(2),
less(3)
}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If this value is greater(2) or less(3), the associated apmExceptionResponsivenessThreshold will be compared to this value and an exception will be created if the responsiveness is greater than the threshold (greater(2)) or less than the threshold (less(3))."

::= { apmExceptionEntry 2 }

apmExceptionResponsivenessThreshold OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The threshold that responsiveness metrics are compared to."

::= { apmExceptionEntry 3 }

apmExceptionUnsuccessfulException OBJECT-TYPE

SYNTAX INTEGER {
off(1),
on(2)
}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If this value is on(2), an exception will be created if a transaction of the associated type is unsuccessful."

::= { apmExceptionEntry 4 }

apmExceptionResponsivenessEvents OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of responsiveness exceptions generated. This counter will be incremented even if no notification was sent due to notifications not being configured or due to exceeding the apmNotificationMaxRate value."

::= { apmExceptionEntry 5 }

apmExceptionUnsuccessfulEvents OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of unsuccessful exceptions generated. This counter will be incremented even if no notification was sent due to notifications not being configured or due to exceeding the apmNotificationMaxRate value."

::= { apmExceptionEntry 6 }

apmExceptionOwner OBJECT-TYPE

SYNTAX OwnerString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The entity that configured this entry and is therefore using the resources assigned to it."

::= { apmExceptionEntry 7 }

apmExceptionStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type of this apmReportControlEntry. If the value of this object is 'permanent', no objects in this row need to be writable."

::= { apmExceptionEntry 8 }

apmExceptionStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this apmExceptionEntry. The only objects in the entry that may be modified while the entry is in the active state are apmExceptionResponsivenessComparison, apmExceptionResponsivenessThreshold and apmExceptionUnsuccessfulException."

::= { apmExceptionEntry 9 }

apmThroughputExceptionMinTime OBJECT-TYPE

SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"Because the responsiveness for throughput-oriented transactions is divided by the elapsed time, it can be very sensitive to short-term performance variations for transactions that take a short period of time. For example, when downloading a very short file, a single dropped packet could double or triple the total response time.

Further, throughput is usually examined for applications that transfer a lot of data, and when doing so it is helpful to conceptualize transaction costs that are proportional to the amount of data separately from those costs that are relatively fixed (i.e., independent of the amount of data). For very short transactions, these fixed transaction costs (handshake, setup time, authentication, round-trip time) may dominate the total response time for the transaction, resulting in throughput measurements that aren't really proportional to the network's, server's and client's combined data throughput capability.

This object controls the minimum number of seconds that an throughput-based transaction must exceed before an exception can be generated for it. If this object is set to zero, then all throughput-based transactions are candidates for exceptions.

The value of this object must persist across reboots."

DEFVAL { 10 }
::= { apmMibObjects 14 }

apmNotificationMaxRate OBJECT-TYPE

SYNTAX Unsigned32
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"The maximum number of notifications that can be generated from this agent by the apmExceptionTable in any 60 second period.

The value of this object must persist across reboots."

DEFVAL { 1 }
::= { apmMibObjects 15 }

-- APM Notifications

apmNotifications OBJECT IDENTIFIER ::= { apm 0 }

apmTransactionResponsivenessAlarm NOTIFICATION-TYPE

OBJECTS { apmExceptionResponsivenessThreshold,
apmTransactionResponsiveness }

STATUS current

DESCRIPTION

"Notification sent when a transaction exceeds a threshold defined in the apmException table. The index of the included apmExceptionResponsivenessThreshold object identifies the apmExceptionEntry that specified the threshold. The apmTransactionResponsiveness variable identifies the actual transaction and its responsiveness.

Agent implementors are urged to include additional data objects in the alarm that may explain the reason for the alarm. It is helpful to include such data in the alarm because it describes the situation at the time the alarm was generated, where polls after the fact may not provide meaningful information. Examples of such information are CPU load, memory utilization, network utilization, and transaction statistics."

::= { apmNotifications 1 }

apmTransactionUnsuccessfulAlarm NOTIFICATION-TYPE

OBJECTS { apmExceptionResponsivenessThreshold }

STATUS current

DESCRIPTION

"Notification sent when a transaction is unsuccessful. The index of the included apmExceptionResponsivenessThreshold object identifies both the type of the transaction that caused this notification as well as the apmExceptionEntry that specified the threshold.

Agent implementors are urged to include additional data objects in the alarm that may explain the reason for the alarm. It is helpful to include such data in the alarm because it describes the situation at the time the alarm was generated, where polls after the fact may not provide meaningful information. Examples of such information are CPU load, memory utilization, network utilization, and transaction statistics."

::= { apmNotifications 2 }

apmCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"Describes the requirements for conformance to the APM MIB"

MODULE -- this module

MANDATORY-GROUPS { apmAppDirGroup, apmReportGroup }

GROUP apmUserDefinedApplicationsGroup

DESCRIPTION

"Implementation of the apmUserDefinedApplicationsGroup is optional."

GROUP apmTransactionGroup

DESCRIPTION

"Implementation of the apmTransactionGroup is optional."

GROUP apmExceptionGroup

DESCRIPTION

"Implementation of the apmExceptionGroup is optional."

GROUP apmNotificationGroup

DESCRIPTION

"Implementation of the apmNotificationGroup is optional."

::= { apmCompliances 1 }

apmAppDirGroup OBJECT-GROUP

OBJECTS { apmAppDirConfig,
apmAppDirResponsivenessBoundary1,
apmAppDirResponsivenessBoundary2,
apmAppDirResponsivenessBoundary3,
apmAppDirResponsivenessBoundary4,
apmAppDirResponsivenessBoundary5,
apmAppDirResponsivenessBoundary6,
apmBucketBoundaryLastChange, apmAppDirID,
apmNameMachineName, apmNameUserName }

STATUS current

DESCRIPTION

"The APM MIB directory of applications and application verbs."

::= { apmGroups 1 }

apmUserDefinedApplicationsGroup OBJECT-GROUP

OBJECTS { apmHttpFilterAppLocalIndex,
apmHttpFilterServerProtocol,
apmHttpFilterServerAddress, apmHttpFilterURLPath,
apmHttpFilterMatchType, apmHttpFilterOwner,
apmHttpFilterStorageType, apmHttpFilterRowStatus,
apmHttpIgnoreUnregisteredURLs, apmHttp4xxIsFailure,
apmUserDefinedAppParentIndex,

```

        apmUserDefinedAppApplication }
STATUS    current
DESCRIPTION
    "Objects used for creating and managing user-defined
    applications."
 ::= { apmGroups 2 }

apmReportGroup OBJECT-GROUP
OBJECTS { apmReportControlDataSource,
          apmReportControlAggregationType,
          apmReportControlInterval,
          apmReportControlRequestedSize,
          apmReportControlGrantedSize,
          apmReportControlRequestedReports,
          apmReportControlGrantedReports,
          apmReportControlStartTime,
          apmReportControlReportNumber,
          apmReportControlDeniedInserts,
          apmReportControlDroppedFrames,
          apmReportControlOwner,
          apmReportControlStorageType,
          apmReportControlStatus,
          apmReportTransactionCount,
          apmReportSuccessfulTransactions,
          apmReportResponsivenessMean,
          apmReportResponsivenessMin,
          apmReportResponsivenessMax,
          apmReportResponsivenessB1,
          apmReportResponsivenessB2,
          apmReportResponsivenessB3,
          apmReportResponsivenessB4,
          apmReportResponsivenessB5,
          apmReportResponsivenessB6,
          apmReportResponsivenessB7 }
STATUS    current
DESCRIPTION
    "The apm report group controls the creation and retrieval of
    reports that aggregate application performance."
 ::= { apmGroups 3 }

apmTransactionGroup OBJECT-GROUP
OBJECTS { apmTransactionResponsiveness,
          apmTransactionAge, apmTransactionSuccess,
          apmTransactionsRequestedHistorySize }
STATUS    current
DESCRIPTION
    "The apm transaction group contains statistics for
    individual transactions."

```

```
::= { apmGroups 4 }
```

apmExceptionGroup OBJECT-GROUP

```
OBJECTS { apmExceptionResponsivenessComparison,
          apmExceptionResponsivenessThreshold,
          apmExceptionUnsuccessfulException,
          apmExceptionResponsivenessEvents,
          apmExceptionUnsuccessfulEvents,
          apmExceptionOwner, apmExceptionStorageType,
          apmExceptionStatus, apmThroughputExceptionMinTime,
          apmNotificationMaxRate }
```

```
STATUS current
```

DESCRIPTION

"The apm exception group causes notifications to be sent whenever transactions are detected that had poor availability or responsiveness."

```
::= { apmGroups 5 }
```

apmNotificationGroup NOTIFICATION-GROUP

```
NOTIFICATIONS { apmTransactionResponsivenessAlarm,
                apmTransactionUnsuccessfulAlarm }
```

```
STATUS current
```

DESCRIPTION

"Notifications sent by an APM MIB agent."

```
::= { apmGroups 6 }
```

END

4. Security Considerations

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

Specifically, most of the read-write and read-create objects in this MIB module may be used to configure an agent to reveal network addresses, application usage information and conversation statistics that may be considered sensitive in some environments.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.

Specifically, this MIB contains network addresses, machines names, user names, application usage information, and conversation statistics. Data of this nature should be considered sensitive and the privacy of the users from whom it was gathered protected. Administrators should restrict read access to this data to specifically authorized individuals or agents that recognize the privacy implications of its release. In situations where read access to this data cannot be restricted, it should not be gathered.

Systems that implement the objects in this MIB module have the capability of measuring the time taken to execute transactions. Depending on the transaction type, some or all of this transaction time may be associated with the time taken to perform security calculations. Such data may help an attacker to use timing attacks to extract secrets from the systems involved in the transactions. See [10] for more information.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [8], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

5. References

5.1. Normative References

- [1] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIV2)", STD 58, RFC 2578, April 1999.
- [2] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Textual Conventions for SMIV2", STD 58, RFC 2579, April 1999.
- [3] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Conformance Statements for SMIV2", STD 58, RFC 2580, April 1999.
- [4] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [5] Waldbusser, S., "Remote Network Monitoring Management Information Base Version 2 using SMIV2", RFC 2021, January 1997.
- [6] Bierman, A., Bucci, C. and R. Iddon, "Remote Network Monitoring MIB Protocol Identifiers", RFC 2895, August 2000.
- [7] Waldbusser, S., "Remote Network Monitoring Management Information Base", STD 59, RFC 2819, May 2000.

5.2. Informative References

- [8] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [9] Berners-Lee, T., Masinter, L. and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, December 1994.
- [10] Boneh, D. and D. Brumley, "Remote timing attacks are practical", Proceedings of 12th USENIX Security Symposium, August 2003.

6. Author's Address

Steven Waldbusser
Email: waldbusser@nextbeacon.com

7. Full Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78 and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights 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; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat 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 implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

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

