

به نام خدا

دانشجویان:

بنیامین مهرجو ۹۶۷۲۵۱۵۲

محمد رحیمی ۹۶۷۲۵۰۳۵

ترجمه ی RFC1155

درس مدیریت شبکه های کامپیوتری

استاد: دکتر موحدی

## Structure and Identification of Management Information for TCP/IP-based Internets

### فهرست مطالب:

1. Status of this Memo (وضعیت این یادداشت) .....	3
2. Introduction (معرفی) .....	3
3. Structure and Identification of Management Information (ساختار و شناسایی اطلاعات مدیریت) .....	4
3.1 Names .....	5
3.1.1 Directory .....	5
3.1.2 Mgmt .....	6
3.1.3 Experimental .....	6
3.1.4 Private .....	6
3.2 Syntax .....	7
3.2.1 Primitive Types .....	7
3.2.1.1 Guidelines for Enumerated INTEGERS (دستورالعمل ها برای اینتجر های شمارش شده) .....	7
3.2.2 Constructor Types .....	7
3.2.3 Defined Types .....	7
3.2.3.1 NetworkAddress .....	7
3.2.3.2 IpAddress .....	7
3.2.3.3 Counter .....	8
3.2.3.4 Gauge .....	8
3.2.3.5 TimeTicks .....	8
3.2.3.6 Opaque .....	8
3.3 Encodings .....	8
4. Managed Objects .....	8
4.1 Guidelines for Object Names .....	9
4.2 Object Types and Instances .....	9
4.3 Macros for Managed Objects .....	12
5. Extensions to the MIB .....	13
6. Definitions .....	14
7. Acknowledgements .....	16
8. References (مراجع) .....	17
9. Security Considerations .....	18
Authors' Addresses .....	18

## ۱- وضعیت این یادداشت

این RFC بازنشری از RFC 1065 است، همراه با یک «وضعیت یادداشت» تغییر یافته به علاوه کمی اصلاحات چاپی کوچک. محتوای فنی این سند نسبت به RFC1065 تغییری نکرده است.

این یادداشت تعاریف رایج برای ساختار و شناسایی اطلاعات مدیریت شبکه‌های TCP/IP را آماده کرده است.

به طور خاص، این یادداشت با یادداشت‌های همراهش که اطلاعات مدیریت را بر اساس پروتکل‌های مدیریت شبکه توصیف می‌کنند، معماری و سیستم عملی و ساده‌ای را برای مدیریت شبکه‌های TCP/IP و به طور خاص اینترنت، بدست می‌دهند.

این یادداشت پروتکل استاندارد را برای جامعه‌ی اینترنتی مشخص می‌کند. وضعیت این یادداشت «توصیه شده» است. انتظار می‌رود پیاده سازی‌های TCP/IP در اینترنت که قابل مدیریت شبکه‌ای هستند، این معیارها را اتخاذ و پیاده‌سازی کنند.

هیئت فعالیت‌های اینترنتی (The Internet Activities Board) توصیه می‌کند که همه‌ی پیاده سازی‌های IP و TCP قابل مدیریت

شبکه‌ای باشند. این موضوع بر پیاده‌سازی Internet MIB (RFC-1156) و حداقل یکی از دو پروتکل توصیه شده‌ی مدیریت -

SNMP (RFC-1157) یا CMOT (RFC-1095) - دلالت دارد. باید متذکر شد که در حال حاضر، SNMP یک استاندارد کامل

اینترنت است و CMOT یک استاندارد پیش نویس است. برای کسب اطلاعات بیشتر در مورد کاربرد و قابلیت اجرای این استاندارد

RFC‌های Host and Gateway Requirements را ببینید.

لطفاً برای کسب اطلاعات فعلی در مورد وضعیت پروتکل‌های استاندارد اینترنت، به آخرین ویرایش RFC IAB Official Protocol Standards رجوع کنید.

توزیع این یادداشت به صورت نامحدود است.

## ۲- معرفی

این یادداشت ساختارها و طرح‌های شناسایی رایج برای توصیف اطلاعات مدیریت استفاده شده در شبکه‌های بر اساس TCP/IP را توصیف می‌کند.

به علاوه، توصیفات یک مدل اطلاعات object نیز برای مدیریت شبکه همراه با یک مجموعه از type های کلی استفاده شده برای توصیف اطلاعات مدیریت، اضافه شده است.

توصیفات رسمی این ساختار با استفاده از Abstract Syntax Notation One (ASN.1) داده شده است.

این یادداشت به صورت گسترده به دغدغه‌های سازمانی و سیاست‌های مدیریتی اهمیت می‌دهد. نه objectهایی که مدیریت شده‌اند را مشخص می‌کند، و نه پروتکل‌هایی که برای مدیریت آن objectها استفاده شده‌اند را.

این مهم‌ها به وسیله دو یادداشت همراه پرداخته شده‌اند: یکی Management Information Base (MIB) را توصیف می‌کند، و دیگری Simple Network Management Protocol (SNMP) را.

این یادداشت بر اساس کارهای انجام شده توسط نیروی ضربتی مهندسی اینترنت (Internet engineering Task Force)، به طور خاص یادداشت کاری با عنوان «ساختار و شناسایی اطلاعات مدیریت برای اینترنت» است. این یادداشت از یک ساختار مشتق شده از یادداشت ذکر شده استفاده می‌کند، اما در یک مورد بسیار قابل توجه متفاوت است: آن یادداشت به طور کامل روی استفاده از مدیریت شبکه‌ی OSI-style تمرکز کرده است. همین‌طور برای استفاده به همراه SNMP مناسب نیست.

این یادداشت قصد رسیدن به دو هدف را دارد: سادگی و توسعه پذیری. هردو از یک موضوع با اهمیت رایج الهام گرفته اند: اگرچه مدیریت شبکه‌های TCP/IP برای مدتی موضوع مطالعه بودند، مولفان گمان نمی‌کنند گستره و عمق چنین فهمی کامل شده باشد. به صورت بی‌پرده تر، ما حس می‌کنیم تجربه‌های گذشته، با اینکه به جامعه بینش می‌دهند اما به سختی قاطع و نهایی هستند. با پروراندن یک SMI

ساده، کمترین تعداد محدودیت‌ها روی روش‌های بالقوه‌ی آینده اعمال میشوند. ثانیاً با پروراندن یک SMI توسعه پذیر بیشترین تعداد روش‌های بالقوه برای آزمایش کردن در دسترس خواهد بود.

باور بر این است که این یادداشت و دو یادداشت همراهش مطابق با راهنماهای تنظیم شده در RFCهای زیر هستند:

RFC 1052 - "IAB Recommendations for Development of Internet Network Management Standard"

RFC 1109 - "Report Of the Second Ad Hoc Network Management Review Group"

به طور خاص ما حس میکنیم که این یادداشت به همراه یادداشتی که پایه‌ی اطلاعات مدیریت را توصیف میکند، بنیادی مستحکم برای مدیریت شبکه‌ی اینترنت به دست میدهند.

### ۳- ساختار و شناسایی اطلاعات مدیریت

objectهای مدیریت شده به وسیله‌ی یک منبع ذخیره‌ی مجازی اطلاعات، تحت عنوان Management Information Base یا MIB در دسترس هستند. Object ها در MIB با استفاده از ASN.1 توصیف شده اند.

هر نوع از object (تحت عنوان object type) یک نام، گرامر و کدگذاری دارد. نام آن به صورت منحصر به فرد تحت عنوان OBJECT IDENTIFIER نمایش داده میشود. یک OBJECT IDENTIFIER درواقع یک نام اختصاص داده شده به صورت مدیریتی است. در مورد سیاست‌های مدیریتی استفاده شده برای اختصاص دادن نام‌ها بعداً در این یادداشت بحث خواهد شد.

گرامر یک object type ساختار داده‌ای انتزاعی مربوط به همان object type را توصیف میکند. برای مثال ساختار یک object type داده شده ممکن است INTEGER یا OCTET STRING باشد. همچنین به طور کلی، ما باید اجازه دهیم هر ساختار ASN.1 برای استفاده در توصیف گرامر یک object type در دسترس باشد، این یادداشت عمده ساختارهای ASN.1 که ممکن است استفاده شوند را محدود کرده است. این محدودیت‌ها منحصراً برای دستیابی به سادگی اعمال شده اند.

کدگذاری یک object type به طور ساده این است که چگونه نمونه‌های آن object type با استفاده از گرامر نوع آن object نمایش داده شده‌اند. به طور ضمنی به تعریف گرامر یک object و کدگذاری اینکه چگونه object در زمان انتقال در شبکه نمایش داده می‌شود، متصل است. این یادداشت استفاده از قوانین پایه‌ی کدگذاری ASN.1 را مشخص میکند.

استفاده از MIB برای مدیریت شبکه یا پروتکل مدیریت شبکه در حوزه‌ی این یادداشت نیست. همانطور که قبلاً اشاره شد این وظایف به یادداشت‌های همراه سپرده شده است. این یادداشت قصد دارد تا محدودیت‌های قرار داده شده بر یادداشت‌های همراه خود را کاهش دهد تا عمومیت افزایش پیدا کند. هرچند در برخی موارد محدودیت‌هایی ایجاد شده است (برای مثال گرامری که ممکن است در زمان توصیف object type در MIB استفاده شود) به منظور تشویق یک سبک مدیریت خاص. ویرایش‌های آینده‌ی این یادداشت ممکن است این محدودیت‌ها را بردارند.

#### ۱.۳- نام‌ها

نام‌ها برای شناسایی objectهای مدیریت شده استفاده میشوند. این یادداشت نام‌هایی را مشخص میکند که به طور طبیعی سلسله مراتبی هستند. مفهوم OBJECT IDENTIFIER برای مدل کردن این فکر استفاده شده است. یک OBJECT IDENTIFIER میتواند برای مقاصدی به غیر از نامگذاری objectهای مدیریت شده استفاده شود. برای مثال، هر استاندارد بین‌المللی به منظور شناسایی یک OBJECT IDENTIFIER اختصاصی دارد. به طور خلاصه، OBJECT IDENTIFIERها برای شناسایی برخی objectها هستند، بدون در نظر گرفتن معانی لغوی که به همراه آن object هستند. برای مثال یک object شبکه، یک سند استاندارد و غیره (...).

یک OBJECT IDENTIFIER ترتیبی است از اعداد صحیح که یک درخت global را پیمایش میکنند. درخت شامل یک ریشه است که به تعدادی گره نام گذاری شده به همراه یال‌ها متصل است. هر گره به نوبه‌ی خود ممکن است دارای فرزندی باشد که خود نام گذاری شده‌اند. در این مورد ممکن است به آن گره یک زیر درخت بگوییم. این فرآیند ممکن است تا یک عمق قراردادی ادامه پیدا کند. در مرکز تعریف OBJECT IDENTIFIER درک این موضوع است که کنترل‌های مدیریتی مفاد اختصاص داده شده به گره ممکن است به منظور پیمایش درخت استفاده شوند. یک برچسب، جفتی است از یک توصیف متنی مختصر و یک عدد صحیح. خود ریشه نام گذاری نشده است، اما حداقل ۳ فرزند به طور مستقیم در زیر خودش دارد: یک گره توسط سازمان استاندارد سازی بین‌المللی مدیریت شده است، که برچسب (1) iso دارد، دیگری توسط کمیته‌ی مشورتی تلفن و تلگراف بین‌المللی مدیریت شده است، که برچسب (0) ccitt دارد و سومی مشترکاً توسط ISO و CCITT مدیریت شده است و برچسب (2) joint-iso-ccitt دارد. سازمان ISO در زیر گره (1) iso زیر درختی را برای استفاده توسط سایر سازمان‌ها تعیین کرده است که برچسب (3) org دارد. ۲ گره از گره‌های فرزند نمایش داده شده، به مؤسسه ملی فناوری و استانداردهای ایالات متحده اختصاص داده شده است. یکی از این زیر درخت‌ها توسط NIST به وزارت دفاع ایالات متحده انتقال داده شده است که برچسب (6) dod دارد. تا زمان تألیف این یادداشت، وزارت دفاع مشخص نکرده است که چگونه از زیر درخت OBJECT IDENTIFIER خود استفاده خواهد کرد. این یادداشت فرض میکند که وزارت دفاع گره‌ای را به جامعه‌ی اینترنت اختصاص خواهد داد، تا به صورت زیر توسط هیئت فعالیت‌های اینترنتی مدیریت شود:

internet OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }

این زیر درخت OBJECT IDENTIFIER اینترنت است که با این پیشوند آغاز میشود:

1.3.6.1.

اکنون این یادداشت به عنوان یک استاندارد تأیید شده توسط IAB، سیاستی که این زیردرخت OBJECT IDENTIFIER بر مبنای آن مدیریت شده است را مشخص میکند. در ابتدا، ۴ گره حضور دارند:

directory OBJECT IDENTIFIER ::= { internet 1 }  
 mgmt OBJECT IDENTIFIER ::= { internet 2 }  
 experimental OBJECT IDENTIFIER ::= { internet 3 }  
 private OBJECT IDENTIFIER ::= { internet 4 }

۱.۱.۳ – فهرست (directory)

زیر درخت (1) directory برای استفاده با یک یادداشت آتی، که توضیح میدهد چگونه OSI Directory ممکن است در اینترنت استفاده شود، رزرو شده است.

۲.۱.۳ – mgmt

زیر درخت (2) mgmt برای شناسایی object هایی استفاده می‌شود که در اسناد تأیید شده‌ی IAB تعریف شده‌اند. مدیریت زیردرخت (2) mgmt توسط IAB به مرجع شماره‌های تخصیص یافته‌ی اینترنت ماکول شده است. وقتی RFC هایی که نسخه‌های جدیدی از استاندارد اینترنت MIB را تعریف میکنند، تأیید شوند، یک OBJECT IDENTIFIER توسط مرجع شماره‌های تخصیص یافته‌ی اینترنت برای شناسایی object هایی که توسط آن یادداشت تعریف شده‌اند، به آن‌ها اختصاص داده میشود. برای مثال RFC ای که استاندارد ابتدایی اینترنت MIB را تعریف میکند ممکن است شماره‌ی سند مدیریت ۱ به آن اختصاص داده شده باشد. این RFC ممکن است از OBJECT IDENTIFIER زیر استفاده کرده باشد:

{ mgmt 1 }

یا

1.3.6.1.2.1

در تعریف استاندارد اینترنت MIB.

تولید نسخه‌های جدید استاندارد اینترنت MIB فرآیندی سخت است. بخش ۵ این یادداشت قوانین مورد استفاده در زمان تعریف نسخه‌ای جدید را توصیف میکند.

۳.۱.۳ – آزمایشی (experimental)

زیر درخت experimental(3) برای شناسایی object های مورد استفاده در آزمایش‌های اینترنت استفاده میشود. مدیریت زیردرخت experimental(3) توسط IAB به مرجع شماره‌های تخصیص یافته‌ی اینترنت موکول شده است.

برای مثال، یک آزمایش کننده ممکن است عدد ۱۷ را دریافت کرده باشد و OBJECT IDENTIFIER را در دسترس داشته باشد.

{ experimental 17 }

یا

1.3.6.1.3.17

به عنوان بخشی از فرآیند تخصیص، مرجع شماره‌های تخصیص یافته‌ی اینترنت ممکن است نیازمندی‌هایی ایجاد کند برای این موضوع که زیر درخت مورد نظر چگونه مورد استفاده قرار گیرد.

۴.۱.۳ – محرمانه (private)

زیر درخت private(4) برای شناسایی object هایی که به صورت یک‌طرفه تعریف شده‌اند استفاده میشود. مدیریت زیر درخت private(4) توسط IAB به مرجع شماره‌های تخصیص یافته‌ی اینترنت موکول شده است. در ابتدا این زیر درخت حداقل یک فرزند دارد:

enterprises OBJECT IDENTIFIER ::= { private 1 }

زیر درخت enterprises(1). علاوه بر موارد دیگری، برای اجازه دادن به اشخاص در زمینه‌ی تهیه‌ی زیر سیستم‌های شبکه‌ای خودشان برای ثبت مدل‌های محصولاتشان، مورد استفاده قرار میگیرد.

در کنار دریافت یک زیر درخت، آن شرکت ممکن است، برای مثال در این زیر درخت MIB Object جدیدی تعریف کند. به علاوه، به شدت توصیه شده است که شرکت به منظور تهیه‌ی یک مکانیزم شناسایی غیر مبهم برای استفاده در پروتکل‌های مدیریت، زیر سیستم‌های شبکه‌ای خود را زیر این زیرسیستم ثبت کند. برای مثال اگر شرکت Flintstones Inc زیرسیستم‌های شبکه‌ای تولید کند، آنگاه آن‌ها میتوانند یک گره در زیر زیردرخت enterprise از مرجع شماره‌های تخصیص یافته‌ی اینترنت درخواست کنند. چنین گره‌ای ممکن است شماره گذاری شده باشد:

1.3.6.1.4.1.42

آنگاه شرکت Flintstones Inc ممکن است Fred Router خود را تحت نام زیر ثبت کند:

1.3.6.1.4.1.42.1.1

۲.۳ – گرامر (Syntax)

گرامر (Syntax) برای تعریف ساختار مربوط به نوع object استفاده میشود. قالب‌های ASN.1 برای تعریف این ساختار استفاده شده اند، هرچند تمامی اصول کلی ASN.1 مجاز نیست.

ObjectSyntax از نوع ASN.1 گرامر متفاوتی را تعریف میکند که ممکن است خود در تعریف یک نوع object استفاده شود.

### ۱.۲.۳ - type های ابتدایی (Primitive Types)

فقط type های ابتدایی (Primitive Types) OBJECT IDENTIFIER, OCTET STRING, INTEGER و NULL که در ASN.1 هستند مجاز میباشند. گاهی اوقات به آنها type های غیرانبوه گفته میشود.

### ۱.۱.۲.۳ - دستورالعملها برای INTEGER های شمارش شده

اگر یک INTEGER شمارش شده به عنوان یک object type لیست شده باشد، آنگاه یک عدد نام گذاری شده که مقدار 0 دارد نباید در لیست شمارشها حضور داشته باشد. استفاده از این مقدار ممنوع شده است.

### ۲.۲.۳ - type های سازنده (Constructor Types)

type سازندهی ASN.1 SEQUENCE مجاز است، و برای تولید هر دو لیست از جداول از آن استفاده میشود.

برای لیستها، گرامر شکل زیر را به خود میگیرد:

SEQUENCE { <type1>, ..., <typeN> }

که هر <type> به یکی از type های ابتدایی که در بالا گفته شد تبدیل میشود. به علاوه، این type های ASN.1 همیشه حضور دارند ( شروط DEFAULT و OPTIONAL در تعریف SEQUENCE ظاهر نمیشوند).

برای جداول، گرامر شکل زیر را به خود میگیرد:

SEQUENCE of <entry>

که <entry> به یک سازندهی لیست تجزیه میشود.

گاهی اوقات به لیستها و جداول type های غیرانبوه گفته میشود.

### ۳.۲.۳ - type های تعریف شده

به علاوه، type های در گسترهی برنامه نیز ممکن است تعریف شوند، مادامی که به یک type ابتدایی مانند لیست، جداول یا سایر type های در گسترهی برنامه تبدیل شوند که به صورت ضمنی در ASN.1 تعریف شده اند. در ابتدا، تعداد کمی type های در گسترهی برنامه تعریف شده اند. یادداشت های آینده بدون شک type های دیگر را تعریف خواهند کرد، به محض اینکه یک توافق و اجماع ایجاد شود.

### ۱.۳.۲.۳ - آدرس شبکه (NetworkAddress)

این انتخاب آدرسی را از یکی از چندین خانوادهی پروتکلها به نمایش میگذارد. در حال حاضر، تنها یک خانوادهی پروتکل، خانوادهی اینترنت، در این انتخاب موجود است.

### ۲.۳.۲.۳ - آدرس های ip (IpAddress)

این type در گسترهی برنامه یک آدرس اینترنتی ۳۲ بیتی را به نمایش میگذارد. این آدرس به عنوان یک OCTET STRING با طول ۴، در ترتیب بایت شبکه نمایش داده میشود.

وقتی این ASN.1 type با استفاده از قوانین پایهی کدگذاری ASN.1 کدگذاری شود، فقط شکل کدگذاری ابتدایی (اساسی) باید استفاده شود.

### ۳.۳.۲.۳ - شمارنده (Counter)

این type در گسترهی برنامه یک عدد صحیح غیر منفی را که به صورت یکی یکی تا زمان رسیدن به یک مقدار بیشینه در هنگام بازگشت و شروع از صفر، افزایش می یابد، نمایش میدهد. این یادداشت یک مقدار بیشینهی  $2^{32}-1$  (4294967295 دهمی) را برای شمارندهها مشخص میکند.

#### 3.2.3.4- معیار (Gauge)

این type در گستره‌ی برنامه یک عدد صحیح غیر منفی را نمایش می‌دهد که ممکن است افزایش یا کاهش پیدا کند، اما در یک مقدار بیشینه نگه‌داشته می‌شود. این یادداشت یک مقدار بیشینه‌ی  $2^{32}-1$  (4294967295) (دهدی) را برای معیارها مشخص می‌کند.

#### 3.2.3.5- نشانه‌های زمان (TimeTicks)

این type در گستره‌ی برنامه یک عدد صحیح غیر منفی را نمایش می‌دهد که زمان را در صدم ثانیه از یک مبدأ خاص می‌شمارد. وقتی که object typeها در MIB که از این ASN.1 type استفاده می‌کند، تعریف شوند، توصیف object type مرجع این مبدأ زمانی را شناسایی می‌کند.

#### ۶.۳.۲.۳- مبهم (Opaque)

این type در گستره‌ی برنامه از قابلیت گذراندن گرامر قراردادی ASN.1 پشتیبانی می‌کند. یک مقدار با استفاده از قوانین پایه‌ی ASN.1 در قالب رشته‌ای از octetها کدگذاری می‌شود. این به نوبه‌ی خود، به عنوان یک OCTET STRING و تحت تأثیر آن مقدار ASN.1 اصلی "double-wrapping"، کدگذاری می‌شود. توجه شود که یک پیاده‌سازی تأیید کننده لازم است فقط قابلیت قبول و شناسایی داده‌های کدگذاری شده به صورت مبهم را داشته باشد. لازم نیست داده را باز و محتوای آن را تفسیر کند. تذکر دیگر این است که با استفاده از type خارجی (EXTERNAL) در ASN.1 کدگذاری‌های غیر از ASN.1 نیز ممکن است در داده‌های کدگذاری شده به صورت مبهم استفاده شود.

#### ۳.۳- کدگذاری‌ها (Encodings)

به محض اینکه یک نمونه از یک object type شناسایی شد، مقدارش ممکن است با اعمال قوانین پایه‌ی کدگذاری ASN.1 روی گرامر آن object type انتقال داده شود.

#### ۴- objectهای مدیریت شده (Managed Objects)

اگرچه هدف این یادداشت تعریف objectهای MIB نیست، اما این یادداشت قالبی را برای استفاده توسط سایر یادداشت‌هایی که این objectها را تعریف می‌کنند، مشخص می‌کند. تعریف یک object type شامل ۵ زمینه است:

OBJECT: یک نام متنی، تحت عنوان OBJECT DESCRIPTOR برای object typeها به همراه OBJECT IDENTIFIER مربوط به خودش.

گرامر (Syntax): گرامر انتزاعی برای object type. این گرامر باید به یک نوع ObjectSyntax از ASN.1 تبدیل شود (که در زیر تعریف شده است).



تعریف(Definition): یک توصیف متنی از معانی object type. از آنجایی که این MIB قرار است در محیط های چند طرفه (چند شخصی) مورد استفاده قرار گیرد، پیاده سازی ها باید اطمینان حاصل کنند که نمونه هایشان از یک object این تعریف را برآورده کند. مثلاً حیاتی است که object ها در همه ی ماشین ها معنی ثابت داشته باشند.

دسترسی(Access): یکی از موارد: فقط خواندنی، قابل خواندن و نوشتن، فقط نوشتنی، یا غیر قابل دسترسی.

وضعیت(Status): یکی از موارد: اجباری، اختیاری، یا منسوخ

یادداشت های آتی همچنین ممکن است زمینه های دیگری را برای object هایی که تعریف میکنند مشخص کنند.

#### ۱.۴ – دستورالعمل های نام های object

هیچ object type ی در MIB استاندارد اینترنت نباید یک زیر شناسه 0 در نامش داشته باشد. این مقدار برای استفاده به همراه الحاقیه های آینده رزرو شده است.

هر OBJECT DESCRIPTOR مربوط به یک object type در MIB استاندارد اینترنت باید یک رشته ی منحصر به فرد اما قابل بخاطر سپاری و چاپ باشد. این موضوع یک زبان رایج برای انسان ها را به منظور استفاده در زمان بحث در مورد MIB توسعه میدهد، و همچنین طرح ریزی جداول را برای رابط های کاربری آسانتر میکند.

#### ۲.۴ – object type ها و نمونه ها

یک object type تعریفی از یک نوع object مدیریت شده است، که به صورت طبیعی اعلانی است.

در مقابل، یک نمونه از object، تعریفی از object ی است که به یک مقدار مشخص محدود و مقید شده است. برای مثال مفهوم یک ورودی در یک جدول مسیریابی ممکن است در MIB تعریف شده باشد. چنین مفهومی به یک object type مربوط است، ورودی های انفرادی در یک جدول مسیریابی خاص که در زمانی مشخص وجود دارد نمونه های object از آن object type هستند. مجموعه ای از object type ها در MIB تعریف شده اند. هر کدام از این چنین object type هایی به صورت منحصر به فرد توسط OBJECT IDENTIFIER خودش نام گذاری شده است و همچنین یک نام متنی دارد، که OBJECT DESCRIPTOR همان object type است. مفادی که توسط آنها به نمونه های object ارجاع داده می شود در MIB تعریف نشده اند. ارجاع به نمونه های object به وسیله ی یک مکانیزم مختص به پروتکل قابل دستیابی است. موافق بودن با SMI برای تعریف این مکانیزم مسئولیت و وظیفه ی هر پروتکل مدیریتی است.

یک object type ممکن است طوری در MIB تعریف شده باشد که نمونه ای از آن object type ، انبوهی از اطلاعاتی که توسط نمونه های تعدادی از object type های «تابعه» به نمایش گذاشته شده اند، را نمایش دهد. برای مثلاً فرض کنیم object type های زیر در MIB تعریف شده اند:

OBJECT:

-----

atIndex { atEntry 1 }

Syntax:

INTEGER

Definition:

عدد رابط برای آدرس فیزیکی

Access:

read-write.

Status:

mandatory.

OBJECT:

-----

atPhysAddress { atEntry 2 }

Syntax:

OCTET STRING

Definition:

آدرس فیزیکی وابسته به رسانه

Rose & McCloghrie [Page 11]

RFC 1155 SMI May 1990

Access:

read-write.

Status:

mandatory.

OBJECT:

-----

atNetAddress { atEntry 3 }

Syntax:

NetworkAddress

Definition:

آدرس شبکه‌ی مربوط به آدرس فیزیکی وابسته به رسانه

address.

Access:

read-write.

Status:

mandatory.

آنگاه، object type چهارمی نیز ممکن است در MIB تعریف شده باشد:

OBJECT:

-----

atEntry { atTable 1 }

Syntax:

AtEntry ::= SEQUENCE {

atIndex

INTEGER,

atPhysAddress

OCTET STRING,

atNetAddress

NetworkAddress

}

Definition:

Access:  
read-write.  
Rose & McCloghrie [Page 12]  
RFC 1155 SMI May 1990  
Status:  
mandatory .

هر نمونه از این object type دربردارنده‌ی اطلاعاتی هستند که توسط نمونه‌های سه object type قبلی به نمایش گذاشته شده اند. به Object type ای که به این روش تعریف شده است یک لیست میگویند.  
به طور مشابه، جداول میتوانند توسط انبوهی از type های لیست شکل یابند. برای مثال یک object type پنجم ممکن است در MIB تعریف شده باشد:

OBJECT:

-----

atTable { at 1 }

Syntax:

SEQUENCE OF AtEntry

Definition:

جدول تفسیر آدرس

Access:  
read-write.  
Status:  
.mandatory

به طوریکه هر نمونه از atTable object اطلاعاتی را دربرداشته باشد که توسط مجموعه‌ای از atEntry object type ها که مجموعاً نمونه‌ی مشخص از atTable Object را تشکیل میدهند، نمایش داده شده است، این یک جدول تفسیر آدرس مشخص است.  
در نظر بگیرید چگونه شخصی ممکن است به یک object ساده در یک جدول اشاره کند (ارجاع دهد). با ادامه دادن مثال قبلی، شخصی ممکن است object type را به این صورت نام گذاری کند:

{ atPhysAddress }

و نمونه‌ی object را با استفاده از مکانیز مختص به پروتکل مشخص کند.

{ atNetAddress } = { internet "10.0.0.52" }

این جفت سازی object type و نمونه‌ی object، میتواند به همه‌ی نمونه‌های atPhysAddress اشاره کند که بخشی از هر ورودی در برخی جداول تفسیر آدرس هستند که مقدار atNetAddress مربوط به آنها { "internet "10.0.0.52" } است.  
برای ادامه با این مثال، در نظر بگیرید چگونه شخصی ممکن است به object انبوهی (لیست) در یک جدول اشاره کند.

با نامگذاری object type به صورت

```
{ internet "10.0.0.52" }.
```

و مشخص کردن نمونه‌ی object با استفاده از یک مکانیزم مختص به پروتکل

```
{ atNetAddress } = { internet "10.0.0.52" }
```

به همه‌ی نمونه‌های یک ورودی در جدولی اشاره میکند که مقدار atNetAddress مربوط به آن { "internet" "10.0.0.52" } است. هر پروتکل مدیریت باید مکانیزمی را برای دسترسی به object type های ساده (غیر انبوه) تهیه کند. هر پروتکل مدیریت مشخص میکند که آیا دسترسی به object type های انبوه را پشتیبانی میکند یا خیر. به علاوه، پروتکل باید مشخص کند کدام نمونه‌ها در زمانی که یک جفت نمونه/نوع object به بیشتر از یک نمونه از یک type اشاره میکند، «بازگشت داده شده» است.

برای برآمدن از عهده‌ی پشتیبانی پروتکل‌های گوناگون مدیریت، همه اطلاعاتی که توسط آن‌ها نمونه‌های یک object type مشخص ممکن است از یکدیگر متمایز شوند، توسط نمونه‌های object type های تعریف شده در MIB نمایش داده شده اند.

۳.۴- ماکروها (Macros) برای object های مدیریت شده (Managed Objects)

به منظور آسان سازی استفاده از ابزارهای پردازش تعریف در MIB، ماکروی (کلان) OBJECT-TYPE ممکن است مورد استفاده قرار گیرد. این ماکرو اجازه میدهد جنبه‌های کلیدی یک object type به روشی رسمی نمایش داده شود.

```
OBJECT-TYPE MACRO ::=
BEGIN
TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
"ACCESS" Access
"STATUS" Status
VALUE NOTATION ::= value (VALUE ObjectName)
Access ::= "read-only"
| "read-write"
| "write-only"
| "not-accessible"
Status ::= "mandatory"
| "optional"
| "obsolete"
END
```

با در نظر گرفتن object type هایی که قبلاً تعریف شدند، ممکن است متصور شویم که تعریف زیر در MIB حضور داشته باشد:

```
atIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
::= { atEntry 1 }
```

```
atPhysAddress OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
STATUS mandatory
```

::= { atEntry 2 }

atNetAddress OBJECT-TYPE

SYNTAX NetworkAddress

ACCESS read-write

STATUS mandatory

::= { atEntry 3 }

atEntry OBJECT-TYPE

SYNTAX AtEntry

ACCESS read-write

STATUS mandatory

::= { atTable 1 }

atTable OBJECT-TYPE

SYNTAX SEQUENCE OF AtEntry

ACCESS read-write

STATUS mandatory

::= { at 1 }

AtEntry ::= SEQUENCE {

atIndex

INTEGER,

atPhysAddress

OCTET STRING,

atNetAddress

NetworkAddress

{

پنج تعریف اول object type ها را توصیف میکنند، و برای مثلاً atIndex OBJECT DESCRIPTOR را به  
OBJECT IDENTIFIER {atEntry 1} ربط میدهند. به علاوه گرامر این Object به صورت (INTEGER) و به  
همراه اجازه‌ی دسترسی (خواندن و نوشتن) و وضعیت (اجباری) تعریف شده است. تعریف ششم یک  
ASN.1 type که AtEntry نامیده می‌شود را توصیف میکند.

#### ۵ – الحاقات به MIB

هر سند استاندارد اینترنت MIB تمام اسناد قبل از خود را منسوخ میکند. قسمت نام، تحت عنوان عقبه  
(دم) با توجه به OBJECT IDENTIFIER زیر:

{ mgmt version-number } {عدد نسخه mgmt}

مورد استفاده برای نام گذاری object ها ، باید بین نسخه‌ها بدون تغییر باقی بماند. نسخه‌های جدید ممکن است:

۱ – object type های قدیمی را منسوخ اعلام کنند (در صورت لزوم)، اما نام‌هایشان را حذف نمیکنند

۲ – افزودن به تعریف یک object type مربوط به یک لیست با استفاده از object type های غیر انبوه، به object type های موجود در  
لیست، یا

۳ – تعریف یک object type کاملاً جدید

نسخه‌های جدید ممکن است شامل موارد زیر نباشند:

## ۱- تغییر معانی هر object تعریف شده‌ی سابق بدون تغییر نام آن object

این قوانین مهم هستند، بدلیل اینکه باعث پشتیبانی آسان‌تری برای نسخه‌های چندگانه‌ی استاندارد اینترنت MIB میشوند. بطور خاص، معانی مربوط به عقبه‌ی یک نام، در میان نسخه‌های مختلف MIB ثابت میماند. بدلیل اینکه نسخه‌های چندگانه‌ی MIB ممکن است در «فضای-عقبه» با هم مشترک باشند، پیاده‌سازی‌هایی که نسخه‌های چندگانه از MIB را پشتیبانی میکنند، میتوانند به طور گسترده‌ای ساده سازی شوند.

هرچند، به عنوان یک پیامد، یک عامل مدیریت ممکن است نمونه‌ای مربوط به والد (در سطح بالاتر) یک object type مورد نظر را بازگرداند. با پیروی از اصل قدرتمندی، در این مورد استثنایی، یک مدیر باید هر اطلاعاتی فرای تعریف object type مورد نظر را نادیده بگیرد. هرچند، اصل قدرتمندی نیازمند این است که یک عمل به رعایت فعالیتهای کنترلی اهمیت دهد: اگر یک نمونه، همان گرامر مشابه object type مورد انتظار خود را ندارد آنگاه آن فعالیت‌های کنترلی باید با شکست مواجه شوند. در هر دو مورد نظارت و کنترل، نام object ای که با یک عملیات بازگردانده می‌شود باید مشابه نامی باشد که توسط یک عملیات درخواست داده میشود.

## 6. Definitions

```
RFC1155-SMI DEFINITIONS ::= BEGIN
```

```
EXPORTS -- EVERYTHING
```

```
internet, directory, mgmt,
```

```
experimental, private, enterprises,
```

```
OBJECT-TYPE, ObjectName, ObjectSyntax, SimpleSyntax,
```

```
ApplicationSyntax, NetworkAddress, IpAddress,
```

```
Counter, Gauge, TimeTicks, Opaque;
```

```
-- the path to the root
```

```
internet OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }
```

```
directory OBJECT IDENTIFIER ::= { internet 1 }
```

```
mgmt OBJECT IDENTIFIER ::= { internet 2 }
```

```
experimental OBJECT IDENTIFIER ::= { internet 3 }
```

```
private OBJECT IDENTIFIER ::= { internet 4 }
```

```
enterprises OBJECT IDENTIFIER ::= { private 1 }
```

```
-- definition of object types
```

```
OBJECT-TYPE MACRO ::=
```

```
BEGIN
```

```
TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
```

```
"ACCESS" Access
```

```
"STATUS" Status
```

```
VALUE NOTATION ::= value (VALUE ObjectName)
```

```
Access ::= "read-only"
```

```
| "read-write"
```

```
| "write-only"
```

```
| "not-accessible"
```

```
Status ::= "mandatory"
```

```
| "optional"
```

```
| "obsolete"
```

END

-- names of objects in the MIB

ObjectName ::=   
OBJECT IDENTIFIER

-- syntax of objects in the MIB

ObjectSyntax ::=   
CHOICE {   
simple   
SimpleSyntax,

توجه شود که ترتیب‌های (SEQUENCES) ساده در اینجا به صورت مستقیم اشاره نشده اند،--

به دلیل ساده نگه داشتن موارد --

(مثلاً برای جلوگیری از سوءاستفاده).--

هرچند type ها در گستره‌ی برنامه که همان ترتیب‌های (SEQUENCES) کدگذاری شده به صورت ضمنی (IMPLICITly) هستند--  
ممکن است در انتخاب زیر ظاهر شوند:--

application-wide   
ApplicationSyntax   
}   
SimpleSyntax ::=   
CHOICE {   
number   
INTEGER,   
string   
OCTET STRING,   
object   
OBJECT IDENTIFIER,   
empty   
NULL   
}

ApplicationSyntax ::=   
CHOICE {   
address   
NetworkAddress,   
counter   
Counter,   
gauge   
Gauge,   
ticks   
TimeTicks,   
arbitrary   
Opaque

-- other application-wide types, as they are

```

-- defined, will be added here
}

-- application-wide types

NetworkAddress ::=
CHOICE {
internet
IpAddress
}
IpAddress ::=
[APPLICATION 0] -- in network-byte order
IMPLICIT OCTET STRING (SIZE (4))
Counter ::=
[APPLICATION 1]
IMPLICIT INTEGER (0..4294967295)
Gauge ::=
[APPLICATION 2]
IMPLICIT INTEGER (0..4294967295)
TimeTicks ::=
[APPLICATION 3]
IMPLICIT INTEGER (0..4294967295)
Opaque ::=
[APPLICATION 4] -- arbitrary ASN.1 value,
IMPLICIT OCTET STRING -- "double-wrapped"
END

```

## 7. Acknowledgements

این یادداشت از سه جمع مشارکت کننده در پیش نویس قبلی تأثیر پذیرفته است:

اول، Lee Labarre از شرکت MITRE Corporation، که به عنوان مؤلف NETMAN SMI، نقشه‌ی راه پایه‌ای SMI را به نمایش گذاشت.

دوم، افراد زیادی که نظرات باارزشی را در مورد این یادداشت قبل از توزیع اولیه‌اش ارائه دادند:

James R. Davin, Proteon  
Mark S. Fedor, NYSERNet  
Craig Partridge, BBN Laboratories  
Martin Lee Schoffstall, Rensselaer Polytechnic Institute  
Wengyik Yeong, NYSERNet



Karl Auerbach, Epilogue Technology  
K. Ramesh Babu, Excelan  
Lawrence Besaw, Hewlett-Packard  
Jeffrey D. Case, University of Tennessee at Knoxville  
James R. Davin, Proteon  
Mark S. Fedor, NYSERNet  
Robb Foster, BBN  
Phill Gross, The MITRE Corporation  
Bent Torp Jensen, Convergent Technology  
Lee Labarre, The MITRE Corporation  
Dan Lynch, Advanced Computing Environments  
Keith McCloghrie, The Wollongong Group  
Dave Mackie, 3Com/Bridge  
Craig Partridge, BBN (chair)  
Jim Robertson, 3Com/Bridge  
Marshall T. Rose, The Wollongong Group  
Greg Satz, cisco  
Martin Lee Schoffstall, Rensselaer Polytechnic Institute  
Lou Steinberg, IBM  
Dean Throop, Data General  
Unni Warriier, Unisys

۸.مراجع:

- [1] Information processing systems - Open Systems Interconnection, "Specification of Abstract Syntax Notation One (ASN.1)", International Organization for Standardization, International Standard 8824, December 1987.
- [2] McCloghrie K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based Internets", RFC 1156, Performance Systems International and Hughes LAN Systems, May 1990.
- [3] Case, J., M. Fedor, M. Schoffstall, and J. Davin, "The Simple Network Management Protocol", RFC 1157, University of Tennessee at Knoxville, Performance Systems International, Performance Systems International, and the MIT Laboratory for Computer Science, May 1990.
- [4] LaBarre, L., "Structure and Identification of Management Information for the Internet", Internet Engineering Task Force working note, Network Information Center, SRI International, Menlo Park, California, April 1988.
- [5] Cerf, V., "IAB Recommendations for the Development of Internet Network Management Standards", RFC 1052, IAB, April 1988.

[6] Cerf, V., "Report of the Second Ad Hoc Network Management Review Group", RFC 1109, IAB, August 1989.

[7] Information processing systems - Open Systems Interconnection, "Specification of Basic Encoding Rules for Abstract Notation One (ASN.1)", International Organization for Standardization, International Standard 8825, December 1987.

#### Security Considerations

Security issues are not discussed in this memo.

#### Authors' Addresses

Marshall T. Rose

PSI, Inc.

PSI California Office

P.O. Box 391776

Mountain View, CA 94039

Phone: (415) 961-3380

EMail: mrose@PSI.COM

Keith McCloghrie

The Wollongong Group

1129 San Antonio Road

Palo Alto, CA 04303

Phone: (415) 962-7160

EMail: sytek!kzm@HPLABS.HP.COM