TABLE OF CONTENTS

1. Database Objects and Attributes .................................................................................................................................................. 2

1.1 Object Representation ............................................................................................................................................................... 2

1.2 Object Templates ......................................................................................................................................................................... 3

1.2.1 Attributes can be mandatory, optional or generated ........................................................................................................ 3

1.2.2 Attributes can have single or multiple values .................................................................................................................. 3

1.2.3 Template Format ............................................................................................................................................................... 4

1.3 Object Types ............................................................................................................................................................................. 4

1.3.1 as-set ................................................................................................................................................................................. 4

1.3.2 aut-num ................................................................................................................................................................................ 6

1.3.3 inet6num ........................................................................................................................................................................... 9

1.3.4 inetnum ............................................................................................................................................................................. 11

1.3.5 inet-rrt ............................................................................................................................................................................. 13

1.3.6 key-cert ............................................................................................................................................................................ 15

1.3.7 mntner ............................................................................................................................................................................. 17

1.3.8 route ............................................................................................................................................................................... 20

1.3.9 route6 ............................................................................................................................................................................ 22

1.3.10 route-set ................................................................................................................................................................... 24

2 Using the ARIN Routing Registry Database ........................................................................................................................................ 26

2.1 Submitting Templates ............................................................................................................................................................... 26

2.1.1 Format of an Email ......................................................................................................................................................... 26

2.2 Creating, Modifying and Deleting an Object ........................................................................................................................................ 27

2.2.1 Object Processing ........................................................................................................................................................... 27

2.2.2 Creating a New Object .................................................................................................................................................. 27

2.2.3 Modifying an Existing Object ....................................................................................................................................... 27

2.2.4 Deleting an Object ...................................................................................................................................................... 28

2.3 Email Update Features ............................................................................................................................................................... 28

2.3.1 MIME Support ............................................................................................................................................................... 28

2.3.2 PGP Support ................................................................................................................................................................. 28

2.3.3 Subject Line Processing ............................................................................................................................................... 29

2.4 Acknowledgements and Notifications ......................................................................................................................................... 30

2.4.1 Acknowledgements ....................................................................................................................................................... 30

2.4.2 Notifications ............................................................................................................................................................... 30

2.5 Protecting Data ....................................................................................................................................................................... 31

2.5.1 Authorization Model ..................................................................................................................................................... 31

2.5.2 Protection of Individual Objects ...................................................................................................................................... 32

3. Querying the Routing Registry Database ........................................................................................................................................... 32

4. Mirroring Requests .................................................................................................................................................................. 33
The ARIN Routing Registry is a database containing routing policy information for network operators within the ARIN service region. ARIN’s Routing Registry improves the ability of network operators to configure and manage their networks. Network operators can submit, maintain, and retrieve routing policy information using ARIN’s Routing Registry and can use this information to configure routing software and hardware.

ARIN’s Routing Registry uses Routing Policy Specification Language (RPSL). RPSL is not a router configuration language, but router configurations can be generated from the abstracted data contained within RPSL objects. For more information on RPSL, please consult RFC 2622, RFC 2650, and RFC 4012.

ARIN’s Routing Registry database is separate from ARIN’s Registration database, which populates data in ARIN’s Whois. Updating information in ARIN’s Routing Registry will NOT update the information in ARIN’s Registration database, or vice versa. Before using the ARIN Routing Registry, you must have a Point of Contact (POC) and Organization record (Org ID) established in ARIN’s Registration Database. Find out more about managing information in ARIN’s Registration database.

Conventions Used in This Document

- We use `<label>` for a placeholder or to indicate syntax.
- We use `[option]` to indicate an optional text or command argument.
- In object templates, we also use square brackets "[ ]" to indicate an attribute type.

### 1. DATABASE OBJECTS AND ATTRIBUTES

#### 1.1 OBJECT REPRESENTATION

The records in ARIN’s Routing Registry are known as "objects". RPSL defines the syntax of database objects (how they are written). An object belongs to one of the object types or classes. We use the two terms - 'type' and 'class' - interchangeably throughout this document.

**Object Types Supported by the ARIN Routing Registry**

<table>
<thead>
<tr>
<th>Object Type (Class Name)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>as-set</td>
<td>specifies attributes for a set of aut-num objects; provides a mapping from the named set to component ASes within the registry</td>
</tr>
<tr>
<td>aut-num</td>
<td>specifies attributes for registering the policies of an AS</td>
</tr>
<tr>
<td>inet6num</td>
<td>specifies attributes for IPv6 allocations and assignments</td>
</tr>
<tr>
<td>inetnum</td>
<td>specifies attributes for IPv4 allocations and assignments</td>
</tr>
<tr>
<td>inet-rtr</td>
<td>defines attributes used to register a router</td>
</tr>
<tr>
<td>key-cert</td>
<td>public key certificate that is stored on the server and may be used with a mntner object for authentication when performing updates</td>
</tr>
<tr>
<td>mntner</td>
<td>attributes defining entities authorized to add, delete, or modify certain objects in the Routing Registry</td>
</tr>
<tr>
<td>route</td>
<td>provides attributes of an IPv4 route advertised on the Internet</td>
</tr>
<tr>
<td>route6</td>
<td>provides attributes of an IPv6 route advertised on the Internet</td>
</tr>
</tbody>
</table>
route-set specifies attributes for a set of route objects; provides a mapping from the named set to component routes within the registry

1.2 OBJECT TEMPLATES

A database object in the routing registry is defined as a list of attribute-value pairs. Object templates show which attributes are allowed in each object type.

You must submit routing registry templates in plain text form. The database software only recognizes the printable ASCII character set. If you use any other character sets, or non-printable characters, it may cause problems and your updates could fail.

Each attribute-value pair must start on a separate line. The software is set up to treat a blank line as the end of an object. This is why you cannot include a completely blank line in the middle of an object.

Attribute names have a precisely defined syntax and only use alpha numeric and the hyphen (-) characters. The attribute name must start at column 0 and must immediately be followed by a colon (:). No spaces or tabs are allowed in between the attribute name and the colon. An example of how this would look is:

mntner: MNT-EXAMPLE

If you enter anything different, you will see an error message and your update will fail.

The first attribute that you need to specify is the one that has the same name as the object type. The database software uses this to identify the object type. Each object is uniquely identified by a set of attribute values. We call this set of attributes the 'class primary key'. For most object types this is the value of the first attribute. In some cases it is a different attribute value or a composite of more than one attribute value. The attributes that make up the 'class primary key' are shown in the object templates.

The value part of the attribute-value pair starts after the colon (:). It can contain some pre-defined keywords, references to other objects and free text. You can refer to other objects by using their 'class primary key' values. These references and the keywords have a precisely defined syntax. If you enter anything different, or if the objects you refer to do not already exist in the database, you will see error messages and your update will fail. The free text has no syntax, but may only contain recognizable characters.

Attribute values may contain spaces and tab characters to help make the information easier to read. Note that spaces work better than tabs, as tabs can display differently on different machines.

1.2.1 ATTRIBUTES CAN BE MANDATORY, OPTIONAL OR GENERATED.

You MUST define 'mandatory attributes' in all instances of an object type. If you do not, then the update will fail.

You can skip 'optional attributes'. However, if you do decide to define them, then both the attribute and its value must be syntactically correct. If they are not, then your update will fail. When you skip an optional attribute, remove it completely from the object. You cannot include the attribute name and leave the value blank.

The database software creates any 'generated attributes'. You can skip generated attributes. Where a user-supplied value is not correct, the software will replace it with a generated value. If the software changes a user-supplied value, it will explain the change in a warning message returned to the user. When you skip a generated attribute, remove it completely from the object. You cannot include the attribute name and leave the value blank.

1.2.2 ATTRIBUTES CAN HAVE SINGLE OR MULTIPLE VALUES.

You can only include a 'single valued attribute' once in an object. That one instance can only have a single value.

You can include a 'multiple valued attribute' many times within an object.
1.2.3 TEMPLATE FORMAT

We use the following definitions in the templates:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[mandatory]</td>
<td>You must include at least one instance of this attribute in an object of the class.</td>
</tr>
<tr>
<td>[optional]</td>
<td>This attribute is optional and you can leave it out completely.</td>
</tr>
<tr>
<td>[generated]</td>
<td>The server automatically generates this attribute and you can leave it out completely.</td>
</tr>
<tr>
<td>[single]</td>
<td>Objects must contain only one instance of this attribute - value pair.</td>
</tr>
<tr>
<td>[multiple]</td>
<td>Objects may contain more than one instance of this attribute.</td>
</tr>
<tr>
<td>[look-up key]</td>
<td>This attribute is indexed.</td>
</tr>
<tr>
<td>[inverse key]</td>
<td>This attribute is in the &quot;reverse&quot; index.</td>
</tr>
<tr>
<td>[primary key]</td>
<td>This attribute is (part of) the class primary key.</td>
</tr>
<tr>
<td>[primary/lookup key]</td>
<td>This attribute is indexed and is also (part of) the class primary key.</td>
</tr>
</tbody>
</table>

In an object template, the first column represents an attribute, the second and third columns specify the type of the attribute and the fourth column tells whether the attribute is (part of) a database key for the object.

1.3 OBJECT TYPES

This section describes the object types (classes) that the ARIN Routing Registry supports.

1.3.1 AS-SET

An as-set object defines a set of aut-num objects.

Here is an as-set object template:

```bash
as-set: [mandatory] [single] [primary/lookup key]
descr: [mandatory] [multiple] [ ]
members: [optional] [multiple] [ ]
mbrs-by-ref: [optional] [multiple] [inverse key]
remarks: [optional] [multiple] [ ]
techn-c: [mandatory] [multiple] [inverse key]
admin-c: [mandatory] [multiple] [inverse key]
notify: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
mnt-lower: [optional] [multiple] [inverse key]
changed: [mandatory] [multiple] [ ]
source: [mandatory] [single] [ ]
```
<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>as-set:</td>
<td>&lt;object name&gt;</td>
<td>A name you give to a group of ASNs. It is an RPSL name that starts with &quot;as-&quot;. The name of an as-set object can be hierarchical. A hierarchical as-set name is a sequence of as-set names and AS Numbers separated by colons. At least one component must be an actual as-set name (i.e. start with &quot;as-&quot;). All the set name components of a hierarchical as-name have to be as-set names.</td>
</tr>
<tr>
<td>descr</td>
<td>&lt;freeform&gt;</td>
<td>A short description of the organization and location where this object is used. The description can have multiple lines. It should contain the full postal address as in the ARIN registration.</td>
</tr>
<tr>
<td>members</td>
<td>list of &lt;as-number&gt; or &lt;as-set-name&gt;</td>
<td>Lists the members of the set. It can be either a list of AS Numbers or other as-set names.</td>
</tr>
<tr>
<td>mbrs-by-ref</td>
<td>list of &lt;mntner-name&gt;</td>
<td>is a list of mntner objects. Any aut-num objects associated with these mntner objects that indicate they're a &quot;member-of&quot; this set will be included in the set. If the value of a &quot;mbrs-by-ref:&quot; attribute is ANY, any aut-num referring to the set is a member of the set. If the &quot;mbrs-by-ref:&quot; attribute is missing, the set is defined explicitly by the &quot;members:&quot; attribute.</td>
</tr>
<tr>
<td>remarks</td>
<td>&lt;freeform&gt;</td>
<td>is free-form text explaining or clarifying the object.</td>
</tr>
<tr>
<td>tech-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for a technical contact. This is someone to be contacted for technical problems such as misconfiguration. Example: tech-c: SJ4-ARIN</td>
</tr>
<tr>
<td>admin-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for an administrative contact. Example: admin-c: JS1001-ARIN</td>
</tr>
<tr>
<td>notify</td>
<td>&lt;email&gt;</td>
<td>is an email address to which the ARIN Routing Registry will send an email when this object is changed.</td>
</tr>
<tr>
<td>mnt-by</td>
<td>list of &lt;mntner-name&gt;</td>
<td>is the maintainier ID of the organization responsible for maintaining this object (typically your own mntner ID).</td>
</tr>
<tr>
<td>mnt-lower</td>
<td>list of &lt;mntner-name&gt;</td>
<td>Specifies the identifier of a registered mntner object used for hierarchical authorization. The authentication method of this mntner object will then be used to authorize the creation of any other as-set objects one level more specific in a hierarchy.</td>
</tr>
</tbody>
</table>
### Attribute Name | <attribute_value(type)> | Description
--- | --- | ---
changed | <email><date> | Specifies who submitted the update, and when the object was updated. The format of the date is YYYYMMDD. Dates in the future are not allowed. If the date is not specified, the database software will add the date when the update was actually processed. There must be at least one "changed:" attribute; if there are more, they must be in ascending date order. This attribute is for the user's own reference. Nothing can be reliably determined by anyone other than the user about the object or its change history by looking at the "changed:" attributes.
Example:
changed: user@example.com 20080130

source | <registry-name> | is the routing registry name. "ARIN" is the value to use here for the ARIN Routing Registry.

Please see [RFC 2622](http://example.com/rfc2622) for a full description of the as-set class and its use.

A sample completed as-set template is below (please do not use this information to fill out your template; this data is included for reference only):

```plaintext
as-set: AS65534:AS-CUSTOMERS
descr: Example, Inc.
descr: 114 Pine Circle
descr: Oakton, NY 11333
descr: US
members: AS65533, AS65532
tech-c: EXAMPLE123-ARIN
admin-c: EXAMPLE456-ARIN
mnt-by: MNT-YOURORGID
changed: user@example.com 20080130
source: ARIN
```

### 1.3.2 AUT-NUM

The aut-num object specifies an Autonomous System (AS) and its routing policies. It refers to a group of IP networks that have a single and clearly defined external routing policy, operated by one or more network operators.

Here is an aut-num object template:

```plaintext
aut-num: [mandatory] [single] [primary/lookup key]
as-name: [mandatory] [single] [ ]
descr: [mandatory] [multiple] [ ]
member-of: [optional] [multiple]
import: [optional] [multiple] [ ]
mp-import: [optional] [multiple] [ ]
export: [optional] [multiple] [ ]
mp-export: [optional] [multiple] [ ]
default: [optional] [multiple] [ ]
mp-default: [optional] [multiple] [ ]
remarks: [optional] [multiple] [ ]
admin-c: [mandatory] [multiple] [inverse key]
tech-c: [mandatory] [multiple] [inverse key]
notify: [optional] [multiple] [inverse key]
mnt-lower: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
```
<table>
<thead>
<tr>
<th>Fields:</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Defined As...</th>
</tr>
</thead>
<tbody>
<tr>
<td>aut-num</td>
<td>&lt;as-number&gt;</td>
<td>is your AS number in the format AS&lt;number&gt;. Leading zeroes (AS0352) are not allowed and will be removed (AS352) by the database software.</td>
</tr>
<tr>
<td>as-name</td>
<td>&lt;object-name&gt;</td>
<td>A descriptive name associated with the AS.</td>
</tr>
<tr>
<td>descr</td>
<td>&lt;freeform&gt;</td>
<td>A short description of the organization and location where this object is used. The description can have multiple lines. It should contain the full postal address as in the ARIN registration.</td>
</tr>
<tr>
<td>member-of</td>
<td>list of &lt;set-name&gt;</td>
<td>identifies a list of as-set objects that this AS number wants to be a member of. This claim, however, should be acknowledged by a respective &quot;mbrs-by-ref:&quot; attribute in the as-set object.</td>
</tr>
<tr>
<td>import</td>
<td>[protocol &lt;protocol-1&gt;] [into &lt;protocol-2&gt;] from &lt;peering-1&gt; [action &lt;action-1&gt;] ... from &lt;peering-N&gt; [action &lt;action-N&gt;] accept &lt;filter&gt;</td>
<td>indicates the AS numbers of the peers from which your AS will receive routing information and describes the routing information you’ll accept. Consult <a href="https://tools.ietf.org/html/rfc2622">RFC 2622</a> for detailed information.</td>
</tr>
<tr>
<td>mp-import</td>
<td>[protocol &lt;protocol-1&gt;] [into &lt;protocol-1&gt;] afi &lt;afi-list&gt; from &lt;peering-1&gt; [action &lt;action-1&gt;] ... from &lt;peering-N&gt; [action &lt;action-N&gt;] accept (&lt;filter&gt;</td>
<td>&lt;filter&gt; except &lt;importexpression&gt;</td>
</tr>
<tr>
<td>export</td>
<td>to &lt;peering-1&gt; [action &lt;action-1&gt;] ... to &lt;peering-N&gt; [action &lt;action-N&gt;] announce &lt;filter&gt;</td>
<td>indicates the AS numbers of the peers to which your AS will provide routing information and describes the routing information you’ll provide. Consult <a href="https://tools.ietf.org/html/rfc2622">RFC 2622</a> for detailed information.</td>
</tr>
<tr>
<td>Fields</td>
<td>&lt;attribute_value(type)&gt;</td>
<td>Defined As...</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td><strong>mp-export</strong></td>
<td>[protocol &lt;protocol-1&gt;] [into &lt;protocol-1&gt;] afi &lt;afi-list&gt; to &lt;peering-1&gt; [action &lt;action-1&gt;] · · · · to &lt;peering-N&gt; [action &lt;action-N&gt;] announce &lt;filter&gt;</td>
<td>Used to specify IPv6 and multicast routing policies.</td>
</tr>
<tr>
<td><strong>default</strong></td>
<td>to &lt;peering&gt; [action &lt;action&gt;] [networks &lt;filter&gt;]</td>
<td>specifies your AS's default routing policy. Consult <a href="https://rfc-editor.org/rfc/rfc2622.html">RFC 2622</a> for detailed information.</td>
</tr>
<tr>
<td><strong>mp-default</strong></td>
<td>to &lt;peering&gt; [action &lt;action&gt;] [networks &lt;filter&gt;]</td>
<td>Used to specify IPv6 and multicast routing policies.</td>
</tr>
<tr>
<td><strong>remarks</strong></td>
<td>&lt;freeform&gt;</td>
<td>is free-form text explaining or clarifying the object.</td>
</tr>
</tbody>
</table>
| **admin-c** | <ARIN POC Handle> | is an ARIN POC handle for an administrative contact.  
Example:  
admin-c: JS1001-ARIN |
| **tech-c** | <ARIN POC Handle> | is an ARIN POC handle for a technical contact. This is someone to be contacted for technical problems such as misconfiguration.  
Example:  
techn-c: SJ4-ARIN |
| **notify** | <email> | is an email address to which the ARIN Routing Registry will send an email when this object is changed. |
| **mnt-lower** | list of <mntner-name> | Specifies the identifier of a registered mntner object used for hierarchical authorization. The authentication method of this mntner object will then be used to authorize the creation of any route and route6 objects one level more specific to this autnum object. |
| **mnt-by** | list of <mntner-name> | is the maintain ID of the organization responsible for maintaining this object (typically your own mntner ID). |
### Fields:

<table>
<thead>
<tr>
<th>changed</th>
<th><code>&lt;email&gt;&lt;date&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Defined As...</td>
<td></td>
</tr>
<tr>
<td>Specifies who submitted the update, and when the object was updated. The format of the date is YYYYMMDD. Dates in the future are not allowed. If the date is not specified, the database software will add the date when the update was actually processed. There must be at least one &quot;changed:&quot; attribute; If there are more, they must be in ascending date order. This attribute is for the user's own reference. Nothing can be reliably determined by anyone other than the user about the object or its change history by looking at the &quot;changed:&quot; attributes. Example: changed: <a href="mailto:user@example.com">user@example.com</a> 20080130</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source</th>
<th><code>&lt;registry-name&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Defined As...</td>
<td></td>
</tr>
<tr>
<td>is the routing registry name. &quot;ARIN&quot; is the value to use here for the ARIN Routing Registry.</td>
<td></td>
</tr>
</tbody>
</table>

Please see [RFC 2622](https://tools.ietf.org/html/rfc2622) for a full description of the aut-num class and its use.

A sample completed aut-num template is below (please do not use this information to fill out your template; this data is included for reference only):

```
aut-num: AS65534
as-name: EXAMPLE-AS
descr: Example, Inc.
descr: 114 Pine Circle
descr: Oakton, NY 11333
descr: US
import: from AS65535 accept ANY
import: from AS65533 accept AS65534
export: to AS65533 announce ANY
export: to AS65535 announce AS2 AS65533
admin-c: EXAMPLE456-ARIN
tech-c: EXAMPLE123-ARIN
mnt-by: MNT-YOURORGID
changed: user@example.com 20080130
source: ARIN
```

### 1.3.3 INET6NUM

An inet6num object contains information on allocations and assignments of IPv6 address space.

Here is an inet6num object template:

```
inet6num: [mandatory] [single] [primary/lookup key]
network: [mandatory] [single] [lookup key]
descr: [mandatory] [multiple] [ ]
country: [mandatory] [multiple] [ ]
admin-c: [mandatory] [multiple] [inverse key]
tech-c: [mandatory] [multiple] [inverse key]
remarks: [optional] [multiple] [ ]
notify: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
mnt-lower: [optional] [multiple] [inverse key]
changed: [mandatory] [multiple] [ ]
source: [mandatory] [single] [ ]
```
<table>
<thead>
<tr>
<th>Fields</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Defined As...</th>
</tr>
</thead>
<tbody>
<tr>
<td>inet6num</td>
<td>&lt;ipv6-address&gt;/&lt;prefix-length&gt;</td>
<td>Specifies a range of IPv6 addresses in prefix notation. The &lt;prefix-length&gt; is an integer in the range from 0 to 128. The range may be a single address. The prefix notation expresses addresses in hexadecimal groups of two bytes separated by colons and with the possible use of shorthand notation for strings of consecutive zeros. Leading zeros from any two-byte group will be removed by the software. In this case a warning message will be returned to the user.</td>
</tr>
<tr>
<td>netname</td>
<td>&lt;netname&gt;</td>
<td>the ARIN-registered netname for this IP network</td>
</tr>
<tr>
<td>descr</td>
<td>&lt;freeform&gt;</td>
<td>A short description of the organization and location where this object is used. The description can have multiple lines. It should contain the full postal address as in the ARIN registration.</td>
</tr>
<tr>
<td>country</td>
<td>&lt;country-code&gt;</td>
<td>gives the two-letter ISO 3166-1 Country Code.</td>
</tr>
<tr>
<td>admin-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for an administrative contact. Example: admin-c: JS1001-ARIN</td>
</tr>
<tr>
<td>tech-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for a technical contact. This is someone to be contacted for technical problems such as misconfiguration. Example: tech-c: SJ4-ARIN</td>
</tr>
<tr>
<td>remarks</td>
<td>&lt;freeform&gt;</td>
<td>is free-form text explaining or clarifying the object.</td>
</tr>
<tr>
<td>notify</td>
<td>&lt;email&gt;</td>
<td>is an email address to which the ARIN Routing Registry will send an email when this object is changed.</td>
</tr>
<tr>
<td>mnt-by</td>
<td>list of &lt;mntner-name&gt;</td>
<td>is the mntner ID of the organization responsible for maintaining this object (typically your own mntner ID).</td>
</tr>
<tr>
<td>mnt-lower</td>
<td>list of &lt;mntner-name&gt;</td>
<td>Specifies the identifier of a registered mntner object used for hierarchical authorization. The authentication method of this mntner object will then be used to authorize the creation of any inet6num and route6 objects one level more specific to this inet6num object.</td>
</tr>
<tr>
<td>changed</td>
<td>&lt;email&gt;&lt;date&gt;</td>
<td>Specifies who submitted the update, and when the object was updated. The format of the date is YYYYMMDD. Dates in the future are not allowed. If the date is not specified, the database software will add the date when the update was actually processed. There must be at least one &quot;changed:&quot; attribute; if there are more, they must be in ascending date order. This attribute is for the user's own reference. Nothing can be reliably determined by anyone other than the user about the object or its change history by looking at the &quot;changed:&quot; attributes. Example: changed: <a href="mailto:user@example.com">user@example.com</a> 20080130</td>
</tr>
</tbody>
</table>
A sample completed `inet6num` template is below (please do not use this information to fill out your template; this data is included for reference only):

```
inetnum: 2001:DB8::/32
netname: EXAMPLE-NET
descr: Example, Inc.
descr: 115 Oak Circle
descr: Pineville, NY 11134
country: US
admin-c: EXAMPLE123-ARIN
tech-c: EXAMPLE456-ARIN
notify: user@example.com
mnt-by: MNT-YOURORGID
changed: user@example.com 20080129
source: ARIN
```

**1.3.4 INETNUM**

An `inetnum` object contains information on allocations and assignments of IPv4 address space.

Here is an `inetnum` object template:

```
itenum: [mandatory] [single] [primary/lookup key]
netname: [mandatory] [single] [lookup key]
descr: [mandatory] [multiple] [ ]
country: [mandatory] [multiple] [ ]
admin-c: [mandatory] [multiple] [inverse key]
tech-c: [mandatory] [multiple] [inverse key]
remarks: [optional] [multiple] [ ]
notify: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
mnt-lower: [optional] [multiple] [inverse key]
changed: [mandatory] [multiple] [ ]
source: [mandatory] [single] [ ]
```
<table>
<thead>
<tr>
<th>Fields</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Defined As...</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for an administrative contact. Example:行政管理联系人: JS1001-ARIN</td>
</tr>
<tr>
<td>tech-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for a technical contact. This is someone to be contacted for technical problems such as misconfiguration. Example:技术联系人: SJ4-ARIN</td>
</tr>
<tr>
<td>remarks</td>
<td>&lt;freeform&gt;</td>
<td>is free-form text explaining or clarifying the object.</td>
</tr>
<tr>
<td>notify</td>
<td>&lt;email&gt;</td>
<td>is an email address to which the ARIN Routing Registry will send an email when this object is changed.</td>
</tr>
<tr>
<td>mnt-by</td>
<td>list of &lt;mntner-name&gt;</td>
<td>is the mntner ID of the organization responsible for maintaining this object (typically your own mntner ID).</td>
</tr>
<tr>
<td>mnt-lower</td>
<td>list of &lt;mntner-name&gt;</td>
<td>Specifies the identifier of a registered mntner object used for hierarchical authorization. The authentication method of this mntner object will then be used to authorize the creation of any inetnum and route objects one level more specific to this inetnum object.</td>
</tr>
<tr>
<td>changed</td>
<td>&lt;email&gt;&lt;date&gt;</td>
<td>Specifies who submitted the update, and when the object was updated. The format of the date is YYYYMMDD. Dates in the future are not allowed. If the date is not specified, the database software will add the date when the update was actually processed. There must be at least one &quot;changed:&quot; attribute; If there are more, they must be in ascending date order. This attribute is for the user’s own reference. Nothing can be reliably determined by anyone other than the user about the object or its change history by looking at the &quot;changed:&quot; attributes. Example:变更: <a href="mailto:user@example.com">user@example.com</a> 20080130</td>
</tr>
<tr>
<td>source</td>
<td>&lt;registry-name&gt;</td>
<td>is the routing registry name. &quot;ARIN&quot; is the value to use here for the ARIN Routing Registry.</td>
</tr>
</tbody>
</table>

A sample completed `inetnum` template is below (please do not use this information to fill out your template; this data is included for reference only):

```
inetnum: 192.0.2.0 – 192.0.2.255
netname: EXAMPLE-NET
descr:  Example, Inc.
descr:  115 Oak Circle
descr:  Pineville, NY 11134
country: US
admin-c: EXAMPLE123-ARIN
tech-c: EXAMPLE456-ARIN
notify: user@example.com
mnt-by: MNT-YOURORGID
changed: user@example.com 20080129
```
1.3.5 INET-RTR

The inet-rtr object specifies routers.

Here is an inet-rtr object template:

```
inet-rtr: [mandatory] [single] [primary/lookup key]
descr: [mandatory] [multiple] [ ]
alias: [optional] [multiple] [ ]
local-as: [mandatory] [single] [inverse key]
ifaddr: [mandatory] [multiple] [lookup key]
interface: [optional] [multiple] [lookup key]
peer: [optional] [multiple] [ ]
mp-peer: [optional] [multiple] [ ]
remarks: [optional] [multiple] [ ]
admin-c: [mandatory] [multiple] [inverse key]
tech-c: [mandatory] [multiple] [inverse key]
notify: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
changed: [mandatory] [multiple] [ ]
source: [mandatory] [single] [ ]
```

<table>
<thead>
<tr>
<th>Fields</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Defined As...</th>
</tr>
</thead>
<tbody>
<tr>
<td>inet-rtr</td>
<td>&lt;domain-name&gt;</td>
<td>is the hostname of the router without a trailing dot.</td>
</tr>
<tr>
<td>descr</td>
<td>&lt;freeform&gt;</td>
<td>A short description of the organization and location where this object is used. The description can have multiple lines. It should contain the full postal address as in the ARIN registration.</td>
</tr>
<tr>
<td>alias</td>
<td>&lt;domain-name&gt;</td>
<td>is an alternate hostname of the router.</td>
</tr>
<tr>
<td>local-as</td>
<td>&lt;as-number&gt;</td>
<td>specifies the AS Number of the AS that owns or operates this router.</td>
</tr>
<tr>
<td>ifaddr</td>
<td>&lt;ipv4-address&gt; masklen &lt;integer&gt; [action &lt;action&gt;]</td>
<td>is the IP address and subnet mask of an interface on the router.</td>
</tr>
<tr>
<td>interface</td>
<td>&lt;ipv4-address&gt; or &lt;ipv6-address&gt; masklen &lt;masklen&gt; &lt;integer&gt; [action &lt;action&gt;] [tunnel &lt;remote-endpoint-address&gt;,&lt;encapsulation&gt;]</td>
<td>extends the “ifaddr:” attribute to allow IPv6 addresses; specifies a multi-protocol interface address within an Internet router, optional action and tunnel definition.</td>
</tr>
<tr>
<td>peer</td>
<td>&lt;protocol&gt;&lt;ipv4-address&gt;&lt;options&gt; &lt;protocol&gt;&lt;inet-rtr-name&gt;&lt;options&gt; &lt;protocol&gt;&lt;rtr-set-name&gt;&lt;options&gt; &lt;protocol&gt;&lt;peering-set-name&gt;&lt;options&gt;</td>
<td>specifies the details of any interior or exterior router peerings.</td>
</tr>
</tbody>
</table>
Fields:  | <attribute_value(type)> | Defined As...
--- | --- | ---
mp-peer | <protocol> afi <afi> <ipv4- or ipv6-address> <options> | extends the "peer:" attribute for IPv6 addresses; specifies the details of any interior or exterior router peerings.
  | <protocol> <inet-rtr-name> <options> |  
  | <protocol> <rtr-set-name> <options> |  
  | <protocol> <peering-set-name> <options> |  

remarks | <freeform> | is free-form text explaining or clarifying the object.

admin-c | <ARIN POC Handle> | is an ARIN POC handle for an administrative contact.
  |  | Example:
  | admin-c: JS1001-ARIN |  

tech-c | <ARIN POC Handle> | is an ARIN POC handle for a technical contact. This is someone to be contacted for technical problems such as misconfiguration.
  |  | Example:
  | tech-c: SJ4-ARIN |  

notify | <email> | is an email address to which the ARIN Routing Registry will send an email when this object is changed.

mnt-by | List of <mntner-name> | is the maintainer ID of the organization responsible for maintaining this object (typically your own mntner ID).

changed | <email><date> | Specifies who submitted the update, and when the object was updated. The format of the date is YYYYMMDD. Dates in the future are not allowed. If the date is not specified, the database software will add the date when the update was actually processed. There must be at least one "changed:" attribute; if there are more, they must be in ascending date order. This attribute is for the user's own reference. Nothing can be reliably determined by anyone other than the user about the object or its change history by looking at the "changed:" attributes.
  |  | Example:
  | changed: user@example.com 20080130 |  

source | <registry-name> | is the routing registry name. "ARIN" is the value to use here for the ARIN Routing Registry.

Please see RFC 2622 for a full description of the inet-rtr class and its use.

A sample completed inet-rtr template is below (please do not use this information to fill out your template; this data is included for reference only):

```
inet-rtr: CRASH.EXAMPLE.NET
descr: Example, Inc.
descr: 115 Oak Circle
descr: Pineville, NY 11134
alias: crasher.example.net
```
alias: test1.example.net
local-as: AS65535
ifaddr: 176.16.1.1 masklen 16
ifaddr: 176.16.10.1 masklen 16
ifaddr: 193.0.0.222 masklen 28
ifaddr: 193.0.0.158 masklen 28
peer: BGP4 176.16.1.3 asno(as5), flap_damp()
admin-c: EXAMPLE123-ARIN
tech-c: EXAMPLE456-ARIN
mnt-by: MNT-YOURORGID
changed: user@example.net 20080129
source: ARIN

1.3.6 KEY-CERT

A key-cert object is a public key certificate that is stored on the server. It is used with a mntner object for authentication when performing updates. ARIN’s Routing Registry supports PGP keys.

Here is a key-cert object template:

key-cert: [mandatory] [single] [primary/lookup key]
method: [generated] [single] [ ]
owner: [generated] [multiple] [ ]
fingerpr: [generated] [single] [inverse key]
certif: [mandatory] [multiple] [ ]
remarks: [optional] [multiple] [ ]
notify: [optional] [multiple] [inverse key]
admin-c: [optional] [multiple] [inverse key]
tech-c: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
changed: [mandatory] [multiple] [ ]
source: [mandatory] [single] [ ]

Fields: <attribute_value(type)> Defined As...

<table>
<thead>
<tr>
<th>key-cert</th>
<th>PGPKEY-&lt;id&gt;</th>
<th>defines the public key stored in the database. &lt;id&gt; is the ID of the PGP public key in 8-digit hexadecimal format without &quot;0x&quot; prefix.</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>&lt;generated&gt;</td>
<td>This attribute is generated by the software and defines the type of the public key. It is not necessary to include it when you create or modify this object. If it is supplied, the software will check the value. If necessary the software will replace the supplied value with generated value. In this case a warning is returned to the user.</td>
</tr>
<tr>
<td>owner</td>
<td>&lt;generated&gt;</td>
<td>This attribute is generated by the software and specifies the owner of the public key. It is not necessary to include it when you create or modify this object. If it is supplied, the software will check the value. If necessary the software will replace the supplied value with generated value. In this case a warning is returned to the user.</td>
</tr>
<tr>
<td>fingerpr</td>
<td>&lt;generated&gt;</td>
<td>This attribute is generated by the software and is a fingerprint of a key certificate. It is not necessary to include it when you create or modify this object. If it is supplied, the software will check the value. If necessary the software will replace the supplied value with generated value. In this case a warning is returned to the user.</td>
</tr>
<tr>
<td>Fields:</td>
<td>attribute_value(type)</td>
<td>Defined As...</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>certif</td>
<td>&lt;public-key&gt;</td>
<td>Contains the public key for a PGP key. The value of the public key should be supplied either using multiple &quot;certif:&quot; attributes, or in one &quot;certif:&quot; attribute. In the first case, this is easily done by exporting the key from your local key ring in ASCII armored format or the certificate from your browser and prepending each line of the key with the string &quot;certif:&quot;. In the second case, line continuation should be used to represent the key. All the lines of the exported key must be included. For PGP, this includes the begin and end markers and the empty line which separates the header from the key body.</td>
</tr>
<tr>
<td>remarks</td>
<td>&lt;freeform&gt;</td>
<td>is free-form text explaining or clarifying the object.</td>
</tr>
<tr>
<td>notify</td>
<td>&lt;email&gt;</td>
<td>is an email address to which the ARIN Routing Registry will send an email when this object is changed.</td>
</tr>
<tr>
<td>admin-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for an administrative contact. Example: admin-c: JS1001-ARIN</td>
</tr>
<tr>
<td>tech-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for a technical contact. This is someone to be contacted for technical problems such as misconfiguration. Example: tech-c: SJ4-ARIN</td>
</tr>
<tr>
<td>mnt-by</td>
<td>List of &lt;mntner-name&gt;</td>
<td>is the maintainer ID of the organization responsible for maintaining this object (typically your own mntner ID).</td>
</tr>
<tr>
<td>changed</td>
<td>&lt;email&gt;&lt;date&gt;</td>
<td>Specifies who submitted the update, and when the object was updated. The format of the date is YYYYMMDD. Dates in the future are not allowed. If the date is not specified, the database software will add the date when the update was actually processed. There must be at least one &quot;changed:&quot; attribute; If there are more, they must be in ascending date order. This attribute is for the user's own reference. Nothing can be reliably determined by anyone other than the user about the object or its change history by looking at the &quot;changed:&quot; attributes. Example: changed: <a href="mailto:user@example.com">user@example.com</a> 20080130</td>
</tr>
<tr>
<td>source</td>
<td>&lt;registry-name&gt;</td>
<td>is the routing registry name. &quot;ARIN&quot; is the value to use here for the ARIN Routing Registry.</td>
</tr>
</tbody>
</table>

A sample completed **key-cert** template is below (please do not use this information to fill out your template; this data is included for reference only):

```
key-cert: PGPKEY-D8CAF4A5
certif: -----BEGIN PGP PUBLIC KEY BLOCK-----
certif: Version: GnuPG v1.4.5 (GNU/Linux) certif:  
certif: mQGiBE4KD00RBAC7kobosjwWdLOecUbeB/UtNWzVQ2h9eI2Y15yifazKGjOUx  
certif: jTzrIfuljG4pz79B8U1kJonNDamj3lzT/6tWo/PVhH0+VxRWN04J0Iv9n0T+cJ  
certif: 9myDjSzZusv0c0Tb9S0B4F0HDN9wNhTJOHJ98zVMZTFd01XkxuAC19QowCgyYLF  
certif: 6c1yetqL+TwKRW2/1rj4aOyj1Yac5l6Pxyh2Kp5nWF/f8/fgxxLb4snDFWFE5ON  
certif: TWzRBc8majEBWkJpA/xhzbCAF50bZ3rOqshTLacX8B89061eHRP6uCKAIwF
```
1.3.7 MNTNER

Objects in ARIN’s Routing Registry are protected by using mntner objects. A mntner object contains the information needed to authorize creation, deletion or modification of any objects that it protects.

Here is a mntner object template:

```
mtnter: [mandatory] [single] [primary/lookup key]
descr: [mandatory] [multiple] [ ]
admin-c: [mandatory] [multiple] [inverse key]
techn-c: [optional] [multiple] [inverse key]
upd-to: [mandatory] [multiple] [inverse key]
mtnt-nfy: [optional] [multiple] [inverse key]
auth: [mandatory] [multiple] [inverse key]
remarks: [optional] [multiple] [ ]
notify: [optional] [multiple] [inverse key]
abuse-mailbox: [optional] [multiple] [inverse key]
mtnt-by: [mandatory] [multiple] [inverse key]
referral-by: [mandatory] [single] [inverse key]
changed: [mandatory] [multiple] [ ]
sourc: [mandatory] [single] [ ]
```

**Fields:**

- **mntner** `<object-name>` is your choice of a unique string identifying your maintainer in the ARIN Routing Registry database. ARIN currently uses the naming convention MNT-[OrgID], where OrgID is your organization ID in ARIN’s Registration database. As an example, if your OrgID is ARIN, your mntner object would be MNT-ARIN.
<table>
<thead>
<tr>
<th>Fields:</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Defined As...</th>
</tr>
</thead>
<tbody>
<tr>
<td>descr</td>
<td>&lt;freeform&gt;</td>
<td>is your organization name.</td>
</tr>
</tbody>
</table>

| admin-c | <ARIN POC Handle> | is an ARIN POC handle for an administrative contact.  
Example:  
admin-c: JS1001-ARIN |

| tech-c  | <ARIN POC Handle> | is an ARIN POC handle for a technical contact. This is someone to be contacted for technical problems such as misconfiguration.  
Example:  
tech-c: SJ4-ARIN |

| upd-to  | <email>             | specifies the email address to be notified when an attempt to update an object protected by this mntner is unsuccessful. |

| mnt-nfy  | <e-mail>            | specifies the email address to be notified when an object protected by this mntner is successfully updated. |

| auth    | <auth-scheme> <scheme-info> | is the method of authorization for your Routing Registry records. ARIN’s Routing Registry currently accepts: |

<table>
<thead>
<tr>
<th>&lt;auth-scheme&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MD5</strong></td>
<td>This scheme is based on the MD5 hash algorithm and provides stronger authentication than CRYP-PW. The authentication information stored in the database is a pass phrase encrypted using md5-crypt algorithm, which is a concatenation of the &quot;$1$&quot; string, the salt, and the 128-bit hash output. Because it uses 8-character salt and an almost unlimited pass phrase, this scheme is more stable against dictionary attacks. However, since the encrypted form is exposed it cannot be considered as a strong form of verification.</td>
</tr>
<tr>
<td><strong>PGPKEY-&lt;id&gt;</strong></td>
<td>Strong scheme of authentication. <code>&lt;id&gt;</code> is the PGP key ID to be used for authentication. This string is the same one that is used in the corresponding key-cert object's &quot;key-cert:&quot; attribute.</td>
</tr>
<tr>
<td><strong>MAIL-FROM</strong></td>
<td>This is the weakest form of authentication. Only users with any of the email addresses specified are authorized to work with objects secured by the maintainer.</td>
</tr>
</tbody>
</table>

Please note that you cannot specify PGPKEY as the authentication method when you are creating a mntner object. You must first create the mntner with MD5 or MAIL-FROM. Use that authentication method to then create your key-cert object. Then you will need to modify the
<table>
<thead>
<tr>
<th>Fields</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Defined As...</th>
</tr>
</thead>
<tbody>
<tr>
<td>mntner</td>
<td></td>
<td><strong>mntner</strong> object, changing the authentication to PGPKEY.</td>
</tr>
<tr>
<td>remarks</td>
<td>&lt;freeform&gt;</td>
<td>is free-form text explaining or clarifying the object.</td>
</tr>
<tr>
<td>notify</td>
<td>&lt;email&gt;</td>
<td>is an email address to which the ARIN Routing Registry will send an email</td>
</tr>
<tr>
<td></td>
<td></td>
<td>when this object is changed.</td>
</tr>
<tr>
<td>abuse-mailbox</td>
<td>&lt;email&gt;</td>
<td>specifies the email address to which abuse complaints should be sent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When this attribute is specified no other email address should be used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for abuse complaints.</td>
</tr>
<tr>
<td>mnt-by</td>
<td>List of &lt;mntner-name&gt;</td>
<td>is the maintainer ID (the value of the &quot;mntner:&quot; in this object) of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>organization responsible for requesting this object.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field should always match the value entered in the mntner: field.</td>
</tr>
<tr>
<td>referral-by</td>
<td>&lt;mntner-name&gt;</td>
<td>Use the same value entered in the &quot;mntner:&quot; field. It is mandatory to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>include this attribute, however, the database software does not currently</td>
</tr>
<tr>
<td></td>
<td></td>
<td>use it.</td>
</tr>
<tr>
<td>changed</td>
<td>&lt;email&gt;&lt;date&gt;</td>
<td>Specifies who submitted the update, and when the object was updated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The format of the date is YYYYMMDD. Dates in the future are not allowed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the date is not specified, the database software will add the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>date when the update was actually processed. There must be at least</td>
</tr>
<tr>
<td></td>
<td></td>
<td>one &quot;changed:&quot; attribute; If there are more, they must be in ascending</td>
</tr>
<tr>
<td></td>
<td></td>
<td>date order. This attribute is for the user’s own reference. Nothing can be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reliably determined by anyone other than the user about the object or its</td>
</tr>
<tr>
<td></td>
<td></td>
<td>change history by looking at the &quot;changed:&quot; attributes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>changed: <a href="mailto:user@example.com">user@example.com</a> 20080130</td>
</tr>
<tr>
<td>source</td>
<td>&lt;registry-name&gt;</td>
<td>is the routing registry name. &quot;ARIN&quot; is the value to use here for the ARIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Routing Registry.</td>
</tr>
</tbody>
</table>

Please see [RFC 2622](https://www.ietf.org/rfc/rfc2622.txt) for the full definition of the mntner class.

A sample completed **mntner** template is below (please do not use this information to fill out your template; this data is included for reference only):

```
mntner:   MNT-YOURORGID
descr:    Example, Inc.
admin-c:  EXAMPLE123-ARIN
tech-c:   EXAMPLE456-ARIN
upd-to:   hostmaster@example.net
mnt-nfy:  hostmaster@example.net
auth:     MAIL-FROM hostmaster@example.net
mnt-by:   MNT-YOURORGID
referral-by: MNT-YOURORGID
changed:  hostmaster@example.net 20080129
source:   ARIN
```
Each interAS route (also known as an interdomain route) originated by an Autonomous System can be specified by using a \textit{route} object for IPv4 addresses.

Here is a \textit{route} object template:

\begin{verbatim}
route: [mandatory] [single] [primary/lookup key]
descr: [mandatory] [multiple] [ ]
origin: [mandatory] [single] [primary/inverse key]
holes: [optional] [multiple] [ ]
member-of: [optional] [multiple] [ ]
inject: [optional] [multiple] [ ]
aggr-mtd: [optional] [single] [ ]
aggr-bndry: [optional] [single] [ ]
export-comps: [optional] [single] [ ]
components: [optional] [single] [ ]
remarks: [optional] [multiple] [ ]
notify: [optional] [multiple] [inverse key]
mnt-lower: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
changed: [mandatory] [multiple] [ ]
source: [mandatory] [single] [ ]
\end{verbatim}

<table>
<thead>
<tr>
<th>Fields</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Defined As...</th>
</tr>
</thead>
<tbody>
<tr>
<td>route</td>
<td>&lt;address-prefix&gt;</td>
<td>is the CIDR prefix of the route to be originated. Together with the &quot;origin:&quot; attribute, constitutes a primary key of the \textit{route} object.</td>
</tr>
<tr>
<td>descr</td>
<td>&lt;freeform&gt;</td>
<td>A short description of the organization and location where this object is used. The description can have multiple lines. It should contain the full postal address as in the ARIN registration.</td>
</tr>
<tr>
<td>origin</td>
<td>&lt;as-number&gt;</td>
<td>is the AS number from which the route will originate.</td>
</tr>
<tr>
<td>holes</td>
<td>list of &lt;address-prefix&gt;</td>
<td>lists the CIDR prefixes of any subnets which are not reachable through the aggregate route.</td>
</tr>
<tr>
<td>member-of</td>
<td>list of &lt;set-name&gt;</td>
<td>lists the \textit{route-set} objects of which this route is a member. This claim, however, should be acknowledged by a respective &quot;mbrs-by-ref:&quot; attribute in the \textit{route-set} object.</td>
</tr>
<tr>
<td>inject</td>
<td>[at &lt;router-expression&gt;] [action &lt;action&gt;] [upon &lt;condition&gt;]</td>
<td>specifies which routers perform the aggregation and when they perform it. In route objects, the router expression can contain only IPv4 expressions.</td>
</tr>
<tr>
<td>aggr-mtd</td>
<td>inbound</td>
<td>outbound &lt;as-expression&gt;</td>
</tr>
<tr>
<td>aggr-bndry</td>
<td>&lt;as-expression&gt;</td>
<td>defines a set of Autonomous Systems, which form the aggregation boundary.</td>
</tr>
<tr>
<td>export-comps</td>
<td>&lt;filter&gt;</td>
<td>is an RPSL filter matching more specifics prefixes that should be exported outside the aggregation boundary.</td>
</tr>
</tbody>
</table>
| Fields      | <attribute_value(type)>       | Defined As...
|------------|------------------------------|-----------------------------------------------------
| components | [ATOMIC] [[<filter>]
|            | [protocol <protocol>
|            | <filter> ...]]              | defines the component routes and protocols used to form the aggregate. `<Protocol>` is a routing protocol name such as BGP4, OSPF or RIP, and `<filter>` is a policy expression. |
| remarks    | <freeform>                   | is free-form text explaining or clarifying the object. |
| notify     | <email>                      | is an email address to which the ARIN Routing Registry will send an email when this object is changed. |
| mnt-lower  | list of <mntner-name>        | Specifies the identifier of a registered mntner object used for hierarchical authorization. The authentication method of this mntner object will then be used to authorize the creation of any route objects one level more specific to this route object. |
| mnt-by     | List of <mntner-name>        | is the maintainer ID of the organization responsible for maintaining this object (typically your own mntner ID). |
| changed    | <email><date>                | Specifies who submitted the update, and when the object was updated. The format of the date is YYYYMMDD. Dates in the future are not allowed. If the date is not specified, the database software will add the date when the update was actually processed. There must be at least one "changed:" attribute; If there are more, they must be in ascending date order. This attribute is for the user’s own reference. Nothing can be reliably determined by anyone other than the user about the object or its change history by looking at the "changed:" attributes. Example:
|            | changed: user@example.com 20080130 | |
| source     | <registry-name>              | is the routing registry name. "ARIN" is the value to use here for the ARIN Routing Registry. |

Please see [RFC 2622](https://www.rfc-editor.org/rfc/rfc2622) for a full description of the route class and its use.

A sample completed route template is below (please do not use this information to fill out your template; this data is included for reference only):

```
route: 192.0.2.0/24
descr: Example, Inc.
descr: 112 Oak Street
descr: Melvindale, California 99999
descr: US
descr: origin: AS65535
mnt-by: MNT-YOURORGID
changed: hostmaster@example.net 20080130
source: ARIN
```
1.3.9 ROUTE6

Each interAS route (also known as an interdomain route) originated by an Autonomous System can be specified by using a route6 object for IPv6 addresses.

Here is a route6 object template:

```
route6: [mandatory] [single] [primary/look-up key]
descr: [mandatory] [multiple] [ ]
origin: [mandatory] [single] [primary/inverse key]
holes: [optional] [multiple] [ ]
member-of: [optional] [multiple] [ ]
inject: [optional] [multiple] [ ]
aggr-mtd: [optional] [single] [ ]
aggr-bndry: [optional] [single] [ ]
export-comps: [optional] [single] [ ]
components: [optional] [single] [ ]
remarks: [optional] [multiple] [ ]
notify: [optional] [multiple] [inverse key]
mnt-lower: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
changed: [mandatory] [multiple] [ ]
source: [mandatory] [single] [ ]
```

### Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Defined As...</th>
</tr>
</thead>
<tbody>
<tr>
<td>route6</td>
<td>&lt;ipv6-address&gt;/&lt;prefix-length&gt;</td>
<td>is the CIDR prefix form of the IPv6 route to be originated. The &lt;prefix length&gt; is an integer in the range from 0 to 128. Together with the &quot;origin:&quot; attribute, constitutes a primary key of the route6 object.</td>
</tr>
<tr>
<td>descr</td>
<td>&lt;freeform&gt;</td>
<td>A short description of the organization and location where this object is used. The description can have multiple lines. It should contain the full postal address as in the ARIN registration.</td>
</tr>
<tr>
<td>origin</td>
<td>&lt;as-number&gt;</td>
<td>is the AS number from which the route will originate.</td>
</tr>
<tr>
<td>holes</td>
<td>list of &lt;ipv6-address-prefix&gt;</td>
<td>lists the CIDR prefixes of any subnets which are not reachable through the aggregate route.</td>
</tr>
<tr>
<td>member-of</td>
<td>list of &lt;set-name&gt;</td>
<td>lists the route-set objects of which this route is a member. This claim, however, should be acknowledged by a respective &quot;mbrs-by-ref:&quot; attribute in the route-set object.</td>
</tr>
<tr>
<td>inject</td>
<td>[at &lt;router-expression&gt;] [action &lt;action&gt;] [upon &lt;condition&gt;]</td>
<td>specifies which routers perform the aggregation and when they perform it. In route6 objects, the router expression can contain only IPv6 expressions.</td>
</tr>
<tr>
<td>aggr-mtd</td>
<td>inbound</td>
<td>outbound [&lt;as-expression&gt;]</td>
</tr>
<tr>
<td>aggr-bndry</td>
<td>&lt;as-expression&gt;</td>
<td>defines a set of Autonomous Systems, which form the aggregation boundary.</td>
</tr>
<tr>
<td>export-comps</td>
<td>&lt;ipv6-filter&gt;</td>
<td>is an RPSL filter matching more specifics prefixes that should be exported outside the aggregation boundary.</td>
</tr>
<tr>
<td>Fields</td>
<td>&lt;attribute_value(type)&gt;</td>
<td>Defined As...</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>components</td>
<td>[ATOMIC] [&lt;filter&gt;]</td>
<td>defines the component routes and protocols used to form the aggregate. &lt;Protocol&gt; is a routing protocol name such as BGP4, OSPF or RIP, and &lt;filter&gt; is a policy expression.</td>
</tr>
<tr>
<td>remarks</td>
<td>&lt;freeform&gt;</td>
<td>is free-form text explaining or clarifying the object.</td>
</tr>
<tr>
<td>notify</td>
<td>&lt;email&gt;</td>
<td>is an email address to which the ARIN Routing Registry will send an email when this object is changed.</td>
</tr>
<tr>
<td>mnt-lower</td>
<td>list of &lt;mntner-name&gt;</td>
<td>Specifies the identifier of a registered mntner object used for hierarchical authorization. The authentication method of this mntner object will then be used to authorize the creation of any route6 objects one level more specific to this route6 object.</td>
</tr>
<tr>
<td>mnt-by</td>
<td>List of &lt;mntner-name&gt;</td>
<td>is the maintainer ID of the organization responsible for maintaining this object (typically your own mntner ID).</td>
</tr>
<tr>
<td>changed</td>
<td>&lt;email&gt;&lt;date&gt;</td>
<td>Specifies who submitted the update, and when the object was updated. The format of the date is YYYYMMDD. Dates in the future are not allowed. If the date is not specified, the database software will add the date when the update was actually processed. There must be at least one &quot;changed:&quot; attribute; If there are more, they must be in ascending date order. This attribute is for the user’s own reference. Nothing can be reliably determined by anyone other than the user about the object or its change history by looking at the &quot;changed:&quot; attributes. Example: changed: <a href="mailto:user@example.com">user@example.com</a> 20080130</td>
</tr>
<tr>
<td>source</td>
<td>&lt;registry-name&gt;</td>
<td>is the routing registry name. &quot;ARIN&quot; is the value to use here for the ARIN Routing Registry.</td>
</tr>
</tbody>
</table>

Please see [RFC 2622](https://tools.ietf.org/html/rfc2622) for a full description of the route6 class and its use.

A sample completed route6 template is below (please do not use this information to fill out your template; this data is included for reference only):

```
route6:  2001:DB8::/32
descr:   Example, Inc.
descr:   112 Oak Street
descr:   Melvindale, California 99999
descr:   US
origin:  AS65535
mnt-by:  MNT-YOURORGID
changed: hostmaster@example.net 20080130
source:  ARIN
```
### 1.3.10 ROUTE-SET

A route-set object is a set of route prefixes and not a set of database route objects. It defines a set of routes that can be represented by route objects or by address prefixes.

Here is a route-set object template:

```plaintext
route-set: [mandatory] [single] [primary/lookup key]
descr: [mandatory] [multiple] [ ]
members: [optional] [multiple] [ ]
mp-members: [optional] [multiple] [ ]
mbrs-by-ref: [optional] [multiple] [inverse key]
remarks: [optional] [multiple] [ ]
tech-c: [mandatory] [multiple] [inverse key]
admin-c: [mandatory] [multiple] [inverse key]
notify: [optional] [multiple] [inverse key]
mnt-by: [mandatory] [multiple] [inverse key]
mnt-lower: [optional] [multiple] [inverse key]
changed: [mandatory] [multiple] [ ]
source: [mandatory] [single] [ ]
```

<table>
<thead>
<tr>
<th>Fields</th>
<th>&lt;attribute_value(type)&gt;</th>
<th>Defined As...</th>
</tr>
</thead>
<tbody>
<tr>
<td>route-set</td>
<td>&lt;object-name&gt;</td>
<td>defines the name of the set. It is an RPSL name that starts with &quot;rs-&quot;.</td>
</tr>
<tr>
<td>descr</td>
<td>&lt;freeform&gt;</td>
<td>A short description of the organization and location where this object is used. The description can have multiple lines. It should contain the full postal address as in the ARIN registration.</td>
</tr>
<tr>
<td>members</td>
<td>list of &lt;address-prefix-range&gt;or &lt;route-set-name&gt;&lt;range-operator&gt;</td>
<td>This is a list of IPv4 address prefixes or other route-set names. The syntax of &lt;route-set-name&gt; is the same as the syntax of &lt;object-name&gt;.</td>
</tr>
<tr>
<td>mp-members</td>
<td>list of &lt;address-prefix-range&gt;or &lt;route-set-name&gt; or &lt;route-set-name&gt;&lt;range-operator&gt;</td>
<td>This is a list of IPv6 address prefixes or other route-set names. The syntax of &lt;route-set-name&gt; is the same as the syntax of &lt;object-name&gt;.</td>
</tr>
<tr>
<td>mbrs-by-ref</td>
<td>list of &lt;mntner-name&gt;</td>
<td>is a list of mntner objects. Any route or route6 objects associated with these mntner objects that indicate they’re a “member-of” this set will be included in the set. If the value of a &quot;mbrs-by-ref:&quot; attribute is ANY, any route/route6 referring to the set is a member of the set. If the &quot;mbrs-by-ref:&quot; attribute is missing, the set is defined explicitly by the &quot;members:&quot; attribute.</td>
</tr>
<tr>
<td>remarks</td>
<td>&lt;freeform&gt;</td>
<td>is free-form text explaining or clarifying the object.</td>
</tr>
</tbody>
</table>
| Fields          | Attribute Value Type | Defined As...
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>tech-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for a technical contact. This is someone to be contacted for technical problems such as misconfiguration. Example: tech-c: SJ4-ARIN</td>
</tr>
<tr>
<td>admin-c</td>
<td>&lt;ARIN POC Handle&gt;</td>
<td>is an ARIN POC handle for an administrative contact. Example: admin-c: JS1001-ARIN</td>
</tr>
<tr>
<td>notify</td>
<td>&lt;email&gt;</td>
<td>is an email address to which the ARIN Routing Registry will send an email when this object is changed.</td>
</tr>
<tr>
<td>mnt-by</td>
<td>List of &lt;mntner-name&gt;</td>
<td>is the maintainer ID of the organization responsible for maintaining this object (typically your own mntner ID).</td>
</tr>
<tr>
<td>changed</td>
<td>&lt;email&gt;&lt;date&gt;</td>
<td>Specifies who submitted the update, and when the object was updated. The format of the date is YYYYMMDD. Dates in the future are not allowed. If the date is not specified, the database software will add the date when the update was actually processed. There must be at least one &quot;changed:&quot; attribute; if there are more, they must be in ascending date order. This attribute is for the user's own reference. Nothing can be reliably determined by anyone other than the user about the object or its change history by looking at the &quot;changed:&quot; attributes. Example: changed: <a href="mailto:user@example.com">user@example.com</a> 20080130</td>
</tr>
<tr>
<td>source</td>
<td>&lt;registry-name&gt;</td>
<td>is the routing registry name. &quot;ARIN&quot; is the value to use here for the ARIN Routing Registry.</td>
</tr>
</tbody>
</table>

Please see [RFC 2622](https://tools.ietf.org/html/rfc2622) for a full description of the route-set class and its use.

Sample completed route-set templates are below (please do not use this information to fill out your template; this data is included for reference only):

```plaintext
route-set:   rs-foo
descr:       Foo networks and customer routes
mbrs-by-ref: MNT-ME, MNT-YOU
tech-c:      EXAMPLE123-ARIN
admin-c:     EXAMPLE456-ARIN
mnt-by:      MNT-YOURORGID
changed:     hostmaster@example.net 20080129
source:      ARIN
route-set:   rs-bar
descr:       Bar networks and customer routes
members:     176.16.0.0/16
mbrs-by-ref: MNT-YOU
tech-c:      EXAMPLE123-ARIN
admin-c:     EXAMPLE456-ARIN
mnt-by:      MNT-YOURORGID
changed:     hostmaster@example.net 20080129
source:      ARIN
route:       192.0.2.0/24
```
2 USING THE ARIN ROUTING REGISTRY DATABASE

2.1 SUBMITTING TEMPLATES

To create, modify or delete an object in ARIN’s Routing Registry database, you need to send an email message containing an object template to rr@arin.net.

Copy and paste the template into the body of a plain-text email. Templates can be found in Section 1.3 ‘Object Types’. Fill out the template and send the email to rr@arin.net.

Make sure to include values for all mandatory fields. You may choose to include values for any or none of the optional fields. Please delete any optional fields that you do not fill out. Please reference Section 1.2 ‘Object Templates’ for more details.

The first object you need to create is the mntner object. Mntner requests are not autoprocessed; your mntner will be created when an ARIN analyst is able to review and process your request.

Please note that you cannot specify PGPKEY as the authentication method when you are creating a mntner object. You must first create the mntner with MD5 or MAIL-FROM authentication. Use that authentication method to then create your key-cert object. Then you will need to modify the mntner object, changing the authentication to PGPKEY.

Once your mntner object is created, you can begin submitting templates for other object types.

2.1.1 FORMAT OF AN EMAIL

The email must be in plain text. The format of the body of an update message may contain one or more objects. The structure of the objects is described in Section 1.1, ‘Object Representation’.

Object definition starts with the class attribute and ends with the first blank line (“\n\n”). You cannot use a blank line in the object, as the software will read this as being the end of the object. We apply a heuristic method to each paragraph of text in the input to determine if it is an object. Any part of the message that is not recognized as an object is ignored. These parts are grouped together at the end of the acknowledgement message.

There are no restrictions on the number of objects in any one email message. But there is a limit on the total size of an update message. It is up to the user to determine how many objects to include in an update message. There are some points to consider.

- Every email update message will result in one acknowledgement email message returned.
- Each object in an update message may generate several notification email messages.
• If you update 1000 objects by sending 1000 emails, each containing one object, this will cause a large number of emails to be returned. Your mail system needs to be able to handle the volume of emails generated in this way. Because of the volume of emails from one source it may also be seen as spam by some mail agents.
• The acknowledgement and notifications are not sent until all the objects in an update message have been processed.
• If you submit one update message containing 1000 objects, no response will be received until all the objects have been processed.
• You need to optimize the number of update messages and number of objects per update to suit your business practices.

2.2 CREATING, MODIFYING AND DELETING AN OBJECT

To create, modify or delete objects, you need to send an email containing one or more objects. Table 1.1, 'Object Types Supported by the ARIN Routing Registry' lists all the object types that are recognized by the database. No other object types can be created. You must use the object templates described in Section 1.3, 'Object Types' to specify the objects. Each instance of an object must contain at least one of each of the mandatory attributes for that object type. An object can contain zero or more instances of each available optional attribute for that object type. One message may contain several objects, even if they each require different operations, for example, creation, modification or deletion.

2.2.1 OBJECT PROCESSING

As a rule, the order of objects in the message is not changed. The database software processes objects one by one, starting with the first recognized object in the message. It is the user's responsibility to order the objects in the message to make sure that all references can be resolved.

When processing each individual object, the software makes many checks including that:

• The syntax of the object is correct.
• The object passes all required authorization checks.
• All references to other objects can be resolved without conflicts.
• The operation does not compromise referential integrity. For example, when an object is to be deleted, the server checks that it is not referenced from any other object in the ARIN Routing Registry Database.

If all checks are successful, the server processes the operation on the object in the ARIN Routing Registry Database. If one of these steps fails, the operation fails for the object as a whole. This is shown in the acknowledgement message and sometimes in notification messages.

Each object in the update message is processed independently of others, so even if one operation fails, the following objects will still be processed. There may, however, be consequences of a previous failed operation.

After the software finishes processing all the objects in the update message, an acknowledgement message is returned to the sender of the original update as specified in the "Reply-to:" field or "From:" field of the email request.

The database server may also send notification messages. See Section 2.4.2, 'Notifications' for more information about this.

2.2.2 CREATING A NEW OBJECT

If the database does not contain an object with the same class primary key as the object in the update message, the server will assume that you want to create it.

2.2.3 MODIFYING AN EXISTING OBJECT

If an object type with the same class primary keys as the object in the update message already exists in the database, the assumed operation is object modification. The server compares the old and new versions of the object and reports a no-operation error if they are identical. When comparing the two versions, white space characters are not considered.
2.4 DELETING AN OBJECT

You can delete an object by adding a "delete:" pseudo attribute to the object.

```
delete: <comment>
```

The software will only accept this request if the object in the message is exactly the same as the one in the database which is to be deleted. When comparing the versions, white space characters are not considered. If you query the database for an object to delete so that you get the exact copy of the object, you should make sure to use the "-B" query flag. Otherwise you will get a filtered object that will not pass the identical check. The delete operation will fail if the object to be deleted is referenced from any other object in the database.

This pseudo attribute applies to one object only in an update message. It must be a part of the object in the update message that is to be deleted. It can be added at any point within the object or immediately after the object. It cannot be placed before the object (this would result in the object not being recognized by the database software as a valid type).

Objects can still be deleted from the database even if they are not syntactically correct. This allows for old objects to be deleted long after the syntax has been changed.

2.3 EMAIL UPDATE FEATURES

This section describes some additional features available for email updates.

2.3.1 MIME SUPPORT

The database software supports MIME. This means that you can cryptographically sign an update message using email agents that attach the signature in a separate MIME part, not in the body of the message. However encryption of the text is not allowed. All update messages must be sent in plain text.

It also allows the definition of scopes of authorization within the message (for example, parts where different passwords apply) and nested signing of messages. This may be necessary under some conditions when updating objects whose authorization must be derived from more than one party.

It is **strongly** recommended to keep MIME encapsulation simple. Complex MIME structures are more likely to generate errors.

The following rules apply when submitting updates using MIME encapsulation:

The software will recognize the following headers and take the appropriate actions:

- multipart/signed
- multipart/alternative
- multipart/mixed
- multipart/unknown
- application/pgp-signature
- text/plain

All other content-types are treated as text/plain.

Each MIME part is treated as a separate message with the following implications (except where a signature part is closely coupled with a text part in which case the two parts are treated together as a message):

- Authorization information is valid only within a single message part

2.3.2 PGP SUPPORT

The database supports PGP signed messages. The following rules apply when submitting updates using this authorization scheme.

- When using MIME encapsulation, a signed portion of an update message should be submitted using multipart/signed composite type. In this case, the first body part contains the update message (which may also be
a MIME encapsulated message), and the second body contains a signature. For a PGP signature, it is encapsulated with application/pgp-signature MIME discrete type.

If one of the signatures fails in a nested signed portion, the whole portion is rejected.

## 2.3.3 SUBJECT LINE PROCESSING

The subject line may have a special meaning in email update messages. Some keywords can be used in certain circumstances. Sometimes users also want to use the subject line for their own reference. All keywords are case insensitive. The available keywords are:

- NEW
- HELP or HOWTO
- DIFF
- KEYWORDS:

One way to use a keyword is to put it in the subject line of the email message, with NO other words present. If any other word is found that is not one of the available keywords (for example, Subject: NEW objects) then none of the words will be treated as keywords. All words in the subject line will be reported in the acknowledgement reply as invalid keywords, along with a WARNING message. In this context it is impossible to know if a word is meant as a keyword or just part of a comment.

Many users often include their own references in the subject line. Using this method it is not possible to also use a keyword. The user's references are reported in the WARNING message as invalid keywords.

Another way to use keywords is with the KEYWORDS: tag. When this tag is found in a subject line all text up to and including the KEYWORDS: tag is ignored. Only the text following this tag is checked against the list of valid keywords. The same rules apply as before: if a word that is not a valid keyword is found after the KEYWORDS: tag, none of the words are actioned as keywords. In this case all the words following the KEYWORDS: tag are reported as invalid keywords in the acknowledgement reply WARNING message.

Adding the KEYWORDS: tag at the end of the Subject: line with no keywords following it means that the user references will not be reported as invalid keywords. This allows you to avoid receiving a WARNING message.

Some examples:

- **Subject: new**
  This is an accepted keyword.

- **Subject: sending my new objects**
  None of these words are accepted as a keyword and all reported in the Warning message.

- **Subject: sending my new objects KEYWORDS: new**
  The last word, "new", is accepted as a keyword, the rest of the line is ignored. No Warning message is generated.

- **Subject: sending my new objects KEYWORDS:**
  No keywords, but no Warning message is generated. The whole line is ignored.

- **Subject: KEYWORDS: sending my new objects**
  None of these words are accepted as a keyword and all reported in the Warning message.

### 2.3.3.1 NEW KEYWORD

Use NEW keyword if you want the database to only accept new objects. In this case all objects found in the update are assumed to be creation operations. If an object already exists in the database, that object will result in an error message in the acknowledgement.
2.3.3.2 HELP (HOWTO) KEYWORD

The HELP keyword causes a help text to be returned in the acknowledgement that contains information about how to query and update the database. When this keyword is used the body of the update message is ignored. HOWTO has the same effect as HELP.

2.3.3.3 DIFF KEYWORD

The DIFF keyword will highlight changes in the notification message between the old and new objects for a modification operation. This is particularly useful when "import:" and "export:" attributes are changed in large aut-num objects. In this case the subject line may contain this:

Subject: changes to aut-num: as1234 KEYWORDS: diff

When the DIFF keyword is used each object in a notification message that has been modified will include a section specifying the changes made, followed by the old version of the full object, and finally the new version of the full object. The output will have the difference listing followed by the old and new objects.

The difference is the standard unix diff, but with one slight change. In the acknowledgement and notification messages three dashes (---) followed by a new line (\n) signifies the start of a section in the message relating to one specific object. This is to make it easy to parse the output and find the start of each object in the message. Note that the standard unix diff also uses --- to separate lines that have changed, so we have replaced --- with === in the diff output presented in the notification messages.

2.4 ACKNOWLEDGEMENTS AND NOTIFICATIONS

2.4.1 ACKNOWLEDGEMENTS

One acknowledgement message (ACK) is returned to the user for each update received. This is split into sections.

The first section shows where the update was received from. This may be a copy of the update email header. This section also includes a summary of the update results, explaining how many objects were recognized by the database software, how many operations were successful and how many failed. The subject line of the ACK message states "SUCCESS" or "FAILED". If the update message contains no objects or any one of the operations fails, the ACK reports "FAILED". Otherwise it reports "SUCCESS". Following this status is the original subject line.

The next section is the "DETAILED EXPLANATION". This is split into three parts. The first part lists all the objects where the operation failed. This will include "Error" messages as well as possible "Info" and "Warning" messages. The next part shows all the operations that were successful. This may include additional "Info" and "Warning" messages with each operation listed. The third part lists all the paragraphs from the update message that were not recognized as objects. Each part is separated in the ACK by a line containing several '~' characters. Within each part the objects and paragraphs are listed in the order they were processed. This is generally the order they appear in the update message. The line before each object contains three '-' characters. This allows for easy parsing by script.

There are many reasons why the operation may fail for the object, including syntax error and authorization failure.

2.4.2 NOTIFICATIONS

There are a number of attributes that may cause notification messages to be generated:

- "notify:"
- "mnt-nfy:"
- "upd-to:"

The "notify:" attribute is an option in most object types, and is used when an object is successfully updated. The "notify:" attribute of the old version of the object is used if the object is being modified or deleted. The "notify:" attribute of the new version of the object is used if the object is being created. If there are multiple "notify:" attributes, an email will be sent to
the addresses contained in all of these, subject to the above conditions on the operation. If the update fails for any reason then no notifications will be sent to any "notify:" email addresses.

The "mnt-nfy:" and "upd-to:" attributes can only be included in mntner objects. The "upd-to:" is mandatory and "mnt-nfy:" is optional. These are used to inform users of successful updates to a maintained object or attempted updates where the authorization failed.

When a maintained object is updated successfully a notification message will be sent to email addresses contained in the "mnt-nfy:" attributes of the mntner objects. If the updated object has one or more "mnt-by:" attributes, notifications will be sent to all the email address listed in all the "mnt-nfy:" attributes in all the referenced mntner objects.

When an update to a maintained object fails the authentication, the notifications are sent to all the email addresses contained in the "upd-to:" attributes. The rules for finding the appropriate "upd-to:" attributes are the same as for the "mnt-nfy:" above.

The format of a notification message is similar to the acknowledgement message. The first section explains why you are being sent this notification. The next section has the email header details showing where the update came from. The final section shows the changes that were contained in the update message, if it was successful. If it failed for authorization reasons, it shows the object for which a change was attempted, but not the actual change details.

A notification message is only sent to a single email address. It will contain all the notification details of objects from an update message that relate to that email address. If the same details need to be sent to two different email addresses, then two separate emails will be generated by the database software. This ensures that when an update contains many objects covered by overlapping notification email addresses, only the appropriate details are sent to each email address.

2.5 PROTECTING DATA

The ARIN Routing Registry Database provides mechanisms to protect objects and control who can make changes to them. In some cases there are also restrictions over who can create certain objects.

- Authentication is the way we determine who is attempting to make a change.
- Authorization is how we decide whether a transaction passing a specific authentication check is allowed to perform a given operation.

Please reference RFC 2725 and RFC 4012 for more information on Routing Policy System Security. Note that these RFCs apply to ARIN's Routing Registry only, and not the ARIN Registration Database.

Different types of objects in the database require different levels of protection. Authentication based on strong encryption is the preferred method, however, this may not always be legally available. For this reason, the server supports multiple authentication methods.

2.5.1 AUTHORIZATION MODEL

The mntner objects serve as a container to hold authentication tokens. A reference to a mntner object within any object defines authorization necessary to perform operations on that object or on a set of related objects. Such reference is provided by means of the "mnt-by:" and "mbrs-by-ref:" attributes.

The mntner object contains one or more "auth:" attributes. Each begins with a keyword identifying the authentication method followed by the authentication information or token needed to enforce that method.

When submitting an update that requires authorization, the authentication information valid for one of the authentication tokens of one of the relevant mntner objects should be supplied. Different methods require different authentication information, as shown below.

Authentication methods currently supported include the following:
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD5-PW</td>
<td>This scheme is based on the MD5 hash algorithm. The authentication information stored in the database is a passphrase encrypted using md5-crypt algorithm, which is a concatenation of the &quot;$1$&quot; string, the salt, and the 128-bit hash output. Because it uses an 8-character salt and an almost unlimited pass phrase, this scheme is quite stable against dictionary attacks. However, since the encrypted form is exposed it cannot be considered as a strong form of authentication. Authentication information is supplied using a &quot;password:&quot; pseudo-attribute. The value of this attribute is a clear-text pass phrase. It can appear anywhere in the body of the message, but not within mail headers. Line continuation is not allowed for this attribute. The attribute and the pass phrase should fit on one line. If you split the pass phrase across multiple lines this will be treated as a syntax error.</td>
</tr>
<tr>
<td>PGPKEY</td>
<td>This is a strong form of authentication. The authentication information is a signature identity pointing to a public key certificate, which is stored in a separate key-cert object. The user is authenticated if the transaction is signed by the corresponding private key. ARIN does not guarantee that a key belongs to any specific entity. Anyone can supply any public keys with any ownership information to the ARIN Routing Registry Database. These keys can be used to protect other objects by checking that the update comes from a user who knows the corresponding secret key.</td>
</tr>
<tr>
<td>MAIL-FROM</td>
<td>This is the weakest for of authentication. MAIL-FROM takes an argument that is a regular expression which covers a set of mail addresses. Only users with any of these mail addresses are authorized to work with objects secured by the corresponding maintainer.</td>
</tr>
</tbody>
</table>

### 2.5.2 PROTECTION OF INDIVIDUAL OBJECTS

Individual objects can be protected with a maintainer object. The maintainer is referenced by the "mnt-by:" attribute in the object. The attribute type is multiple, so several maintner objects can be used to protect one object.

Only those maintner objects referenced by the "mnt-by:" attributes are authorized to modify or delete the object. Note that authentication checks are logically OR-ed (e.g. A or B or C). If the information required by at least one authentication token from one maintner object is supplied, the operation will be authorized. That means that object protection is as weak as the weakest authentication method used in the maintner objects referenced by an object.

When the "mnt-by:" attribute is added to an object for the first time (as part of object creation or modification), the operation should pass authentication checks for at least one of the maintner objects referenced by one of the "mnt-by:" attributes.

If the operation is a modification and the old object already has one or more "mnt-by:" attributes, then one of the maintner objects referenced in one of the "mnt-by:" attributes in the old object must authenticate the change. If the old object does not have any "mnt-by:" attributes, then one of the maintner objects referenced in one of the "mnt-by:" attributes in the new object must authenticate the change. All new objects must have at least one "mnt-by:".

If the operation is a creation then one of the maintner objects referenced in one of the "mnt-by:" attributes in the new object must authenticate the change.

### 3. QUERYING THE ROUTING REGISTRY DATABASE

You can query ARIN’s Routing Registry using a WHOIS client. To query the Routing Registry database using most command-line WHOIS clients, use the following command:

```
whois -h rr.arin.net <object>
```

For a list of instructions on querying, use the following command:

```
whois -h rr.arin.net help
```

If you need help installing or using a WHOIS client, please consult your system administrator.
4. MIRRORING REQUESTS

ARIN's routing registry server mirrors other RPSL databases. In order to be mirrored, a site must show that its routing registry data:

1. is actively maintained
2. has been updated within the past 30 days
3. is publicly available
4. expresses routing policies of use or interest to ARIN's region

The representative of the organization submitting the mirroring request must read and agree to the terms and conditions of the "Routing Registry" section of the ARIN Number Resource Policy Manual (NPRM).

ARIN may cease mirroring if a database fails to meet the criteria listed above. Final judgment on all mirroring requests will be made by ARIN.

To request that ARIN mirror your site’s RPSL database, please send an email to rtreg@arin.net with "MIRRORING REQUEST" in the subject line.