COMPASS
Flow Analyze

Compass

by Calix

Flow Analyze 2.1
Database API and Schema

Rev. 3
January 2014

About this Document

This document describes the interfaces for extracting information from the **Compass Flow Analyze** (FA) application, as well as the FA database schema.

Revision History

Date	Change	Revision
02/2013	Initial Release	1
07/2013	Add 6.3EndPointQuery and 6.5.2 EndpointQuery.wsdl	2
1/2014	Add 'Mapping Detail' and updated Usage.wsdl	3

Table of Contents

1. Introduction

This document describes the database APIs to enable client programs to remotely query the database. It also describes the database schema for those who intend to export the database and data mine remotely.

The database itself is implemented using MySQL. Details of how to export the database and import into your own MySQL are provided in a separate document.

2. Database Interfaces

2.1 top_applications_for_endpoint
2.2 top_applications_for_location
2.3 top_applications_for_interface
2.4 top_endpoints_for_endpoint
2.5 top_endpoints_for_location
2.6 top_endpoints_for_interface
2.7 top_endpoints_for_unmapped_port
2.8 top_endpoints_for_application
2.9 top_locations_for_application
2.10 top_locations_for_interface
2.11 top_locations_for_location
2.12 top_interfaces_for_interface
2.13 top_interfaces_for_endpoint
2.14 top_interfaces_for_application
2.15 top_interfaces_for_location
2.16 overall_top_endpoints
2.17 overall_top_applications
2.18 overall_top_unmapped_IPs
2.19 overall_top_unmapped_ports
2.20 overall_top_locations
2.21 overall_top_interfaces
2.22 get_endpoints_name_list
2.23 get_applications_name_list
2.24 get_locations_name_list
2.25 get_links_name_list
2.26 get_port_list

The client program can use these APIs by sending requests to the CGI based SOAP server through Tomcat running on port 443 on the Flow Analyze Server. On the server side, these APIs are implemented using the Perl module "SoapApiBackend". When using the API on the server, the client program needs to use "SoapApiBackend" as the service module. SOAP server in our implementation uses HTTP as the transport service.

For the APIs in which a time period is required, the minimum duration is 1 hour, and the time period must start no more than 12 weeks prior to the time the API is submitted.

Top N Definition
Maximum value for N is 100. The output for API may contain up to first 100 items from candidate list sorted by bps.

2.1 top_applications_for_endpoint

List top applications interacting with particular endpoint.

- **Syntax**
  top_applications_for_endpoint(`"<endpoint_name>", "<start_time>", "<end_time>")`

- **Parameters**
  `endpoint_name:` complete name of the endpoint, e.g., "`mail.calix.com`"
  `start_time:` starting time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`
  `end_time:` ending time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<app_name>"`
  `dir:` direction of this result (IN or OUT)
  `usage:` usage (in bytes) for this top application in this direction during the specified time period
  `app_name:` name of this top application

2.2 top_applications_for_location

List top applications interacting with a particular location

- **Syntax**
  top_applications_for_location`(“<location_name>","<start_time>", "<end_time>")`

- **Parameters**
  `location_name:` complete name of the location, e.g., `"Location_16_10_5_0_0".`
  `start_time:` starting time in `"<date>T<time>"` format

("yyyy:mm:ddThh:mm:ss")
end_time: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format "<dir>|<usage>|<app_name>"
  dir: direction of this result (IN or OUT)
  usage: usage (in bytes) for this top application in this direction during the specified time period
  app_name: name of this top application

2.3 top_applications_for_interface

List top applications interacting with a particular interface.

- **Syntax:**
  top_applications_for_interface(“<interface_name>","<start_time>","<end_time>")

- **Parameters**
  interface_name: complete name of the interface, e.g., "10_1_0_10_2"
  start_time: starting time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")
  end_time: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format "<dir>|<usage>|<app_name>"
  dir: direction of this result (IN or OUT).
  usage: usage (in interface utilization percentage) for this top application in this direction during the specified time period
  app_name: name of this top application

2. 4 top_endpoints_for_endpoint

List top endpoints interacting with a particular endpoint.

- **Syntax**
  top_endpoints_for_endpoint(“<endpoint_name>","<start_time>","<end_time>")

- **Parameters**

`endpoint_name:` complete name of the endpoint, e.g., "mail.calix.com."
`start_time:` starting time in `"<date>T<time>"` format (`"yyyy:mm:ddThh:mm:ss"`)
`end_time:` ending time in `"<date>T<time>"` format (`"yyyy:mm:ddThh:mm:ss"`)

- **Synopsis**

  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<endpt_name>"`
  `dir:` direction of this result (IN or OUT)
  `usage:` usage (in bytes) for this top endpoint in this direction during the specified time period
  `endpt_name:` name of this top endpoint

2.5. top_endpoints_for_location

List top endpoints interacting with a particular location

- **Syntax**
  top_endpoints_for_location`(“<location_name>","<start_time>", "<end_time>")`

- **Parameters**
  `location_name:` complete name of the location, e.g., `"Location_16_10_5_0_0"`
  `start_time:` starting time in `"<date>T<time>"` format (`"yyyy:mm:ddThh:mm:ss"`)
  `end_time:` ending time in `"<date>T<time>"` format (`"yyyy:mm:ddThh:mm:ss"`)

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<endpt_name>"`
  `dir:` direction of this result (IN or OUT)
  `usage:` usage (in bytes) for this top endpoint in this direction during the specified time period
  `endpt_name:` name of this top endpoint

2.6 top_endpoints_for_interface

List top endpoints interacting with a particular interface.

- **Syntax**
  top_endpoints_for_interface(“<interface_name>","<start_time>", "<end_time>")

- **Parameters**
  interface_name: complete name of the interface, e.g., "10_1_0_10_2"
  start_time: starting time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")
  end_time: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format "<dir>|<usage>|<endpt_name>"
  dir: direction of this result (IN or OUT)
  usage: usage (in interface utilization percentage) for this top endpoint in this direction during the specified time period
  endpt_name: name of this top endpoint

2.7 top_endpoints_for_unmapped_port

List top endpoints interacting with an unmapped port.

- **Syntax**
  top_endpoints_for_unmapped_port("port/proto","from_date", "to_date")

- **Parameters**
  port/proto: port number with protocol, e.g., "10121/TCP"
  start_time: starting time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")
  end_time: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format "<dir>|<usage>|<endpt_name>"
  dir: direction of this result (IN or OUT)
  usage: usage (in bytes) for this top endpoint in this direction during the specified time period
  endpt_name: name of this top endpoint

2.8 top_endpoints_for_application

List top endpoints interacting with a particular application.

- **Syntax**
  top_endpoints_for_application(“<application_name>","<start_time>"," <end_time>”)

- **Parameters**
  application_name: complete name of the application, e.g., "dns"
  start_time: starting time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")
  end_time: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format "<dir>|<usage>|<endpt_name>"
  dir: direction of this result (IN or OUT)
  usage: usage (in bytes) for this top endpoint in this direction during the specified time period
  endpt_name: name of this top endpoint

2.9 top_locations_for_application

List top locations interacting with a particular application.

- **Syntax**
  top_locations_for_application(“<application_name>","<start_time>", "<end_time>”)

- **Parameters**
  application_name: complete name of the application, e.g., "dns"
  start_time: starting time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")
  end_time: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format "<dir>|<usage>|<loc_name>"
  dir: direction of this result (IN or OUT)
  usage: usage (in bytes) for this top location in this direction during the specified time period
  loc_name: name of this top location

2.10 top_locations_for_interface

List top locations interacting with a particular interface.

- **Syntax**
  top_locations_for_interface(“<interface_name>","<start_time>", "<end_time>”)

- **Parameters**
  interface_name: complete name of the interface, e.g., "10_1_0_10_2"
  start_time: starting time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")
  end_time: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format "<dir>|<usage>|<loc_name>"
  dir: direction of this result (IN or OUT)
  usage: usage (in interface utilization percentage) for this top location in this direction during the specified time period
  loc_name: name of this top location

2.11 top_locations_for_location

List top locations interacting with a particular location.

- **Syntax**
  top_locations_for_location(“<location_name>","<start_time>","<end_time>”)

- **Parameters**
  location_name: complete name of the location, e.g., "Location_16_10_5_0_0"
  start_time: starting time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")
  end_time: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format "<dir>|<usage>|<loc_name>"
  dir: direction of this result (IN or OUT)
  usage: usage (in bytes) for this top location in this direction during the specified

time period
`loc_name:` name of this top location

2.12 top_interfaces_for_interface

List top interfaces interacting with a particular interface.

- **Syntax**
  top_interfaces_for_interface(“<interface_name>","<start_time>", "<end_time>”)

- **Parameters**
  interface_name: complete name of the interface, e.g., "10_1_0_10_2"
  `start_time:` starting time in `"<date>T<time>"` format (`"yyyy:mm:ddThh:mm:ss"`)
  `end_time:` ending time in `"<date>T<time>"` format (`"yyyy:mm:ddThh:mm:ss"`)

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<inf_name>"`
  `dir:` direction of this result (IN or OUT)
  `usage:` usage (in interface utilization percentage) for this top interface in this direction during the specified time period
  `inf_name:` name of this top interface.

2.13 top_interfaces_for_endpoint

List top interfaces interacting with a particular endpoint.

- **Syntax**
  top_interfaces_for_endpoint(“<endpoint_name>","<start_time>", "<end_time>”)

- **Parameters**
  endpoint_name: complete name of the endpoint, e.g., "mail.calix.com"
  `start_time:` starting time in `"<date>T<time>"` format (`"yyyy:mm:ddThh:mm:ss"`)
  `end_time:` ending time in `"<date>T<time>"` format (`"yyyy:mm:ddThh:mm:ss"`)

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<inf_name>"`

`dir`: direction of this result (IN or OUT)
`usage`: usage (in interface utilization percentage) for this top interface in this direction during the specified time period
`inf_name`: name of this top interface

2.14 top_interfaces_for_application

List top interfaces interacting with a particular application.

- **Syntax**
  top_interfaces_for_application(“<application_name>","<start_time>","<end_time>”)

- **Parameters**
  `application_name`: complete name of the application, e.g., "dns"
  `start_time`: starting time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")
  `end_time`: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format "<dir>|<usage>|<inf_name>"
  `dir`: direction of this result (IN or OUT)
  `usage`: usage (in interface utilization percentage) for this top interface in this direction during the specified time period
  `inf_name`: name of this top interface

2.15 top_interfaces_for_location

List top interfaces interacting with a particular location.

- **Syntax**
  top_interfaces_for_location(“<location_name>","<start_time>", "<end_time>”)

- **Parameters**
  location_name: complete name of the location, e.g., "Location_16_10_5_0_0"
  `start_time`: starting time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")
  `end_time`: ending time in "<date>T<time>" format ("yyyy:mm:ddThh:mm:ss")

- **Synopsis**

Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<inf_name>"`
`dir:` direction of this result (IN or OUT)
`usage:` usage (in interface utilization percentage) for this top interface in this direction during the specified time period
`inf_name:` name of this top interface

2.16 overall_top_endpoints

List top endpoints across the whole network.

- **Syntax**
  overall_top_endpoints`("<start_time>","<end_time>”)`

- **Parameters**
  `start_time:` starting time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`
  `end_time:` ending time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<endpt_name>"`
  `dir:` direction of this result (IN or OUT)
  `usage:` usage (in bytes) for this top endpoint in this direction during the specified time period
  `endpt_name:` name of this top endpoint

2.17 overall_top_applications

List top applications across the whole network.

- **Syntax**
  overall_top_applications`("<start_time>","<end_time>”)`

- **Parameters**
  `start_time:` starting time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`
  `end_time:` ending time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<app_name>"`

`dir`: direction of this result (IN or OUT)
`usage`: usage (in bytes) for this top application in this direction during the specified time period
`app_name`: name of this top application

2.18 overall_top_unmapped_IPs

List top unmapped IPs across the whole network.

- **Syntax**
  overall_top_unmapped_IPs`("<start_time>","<end_time>”)`

- **Parameters**
  `start_time`: starting time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`
  `end_time`: ending time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<IP>"`
  `dir`: direction of this result (IN or OUT)
  `usage`: usage (in bytes) for this top IP in this direction during the specified time period
  `IP`: This top IP address

2.19 overall_top_unmapped_ports

List top unmapped ports across the whole network.

- **Syntax**
  overall_top_unmapped_ports`("<start_time>","<end_time>”)`

- **Parameters**
  `start_time`: starting time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`
  `end_time`: ending time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<port/proto>"`
  `dir`: direction of this result (IN or OUT)

`usage:` usage (in bytes) for this top port/proto in this direction during the specified time period
`port/proto:` This top port number with protocol

2.20 overall_top_locations

List top locations across the whole network.

- **Syntax**
  overall_top_locations`("<start_time>","<end_time>”)`

- **Parameters**
  `start_time:` starting time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`
  `end_time:` ending time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<loc_name>"`
  `dir:` direction of this result (IN or OUT)
  `usage:` usage (in bytes) for this top location in this direction during the specified time period
  `loc_name:` name of this top location

2. 21 overall_top_interfaces

List top interfaces across the whole network.

- **Syntax**
  overall_top_interfaces`("<start_time>","<end_time>”)`

- **Parameters**
  `start_time:` starting time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`
  `end_time:` ending time in `"<date>T<time>"` format `("yyyy:mm:ddThh:mm:ss")`

- **Synopsis**
  Returns a reference to the query result for iterating. Each iterated result is a string with format `"<dir>|<usage>|<inf_name>"`
  `dir:` direction of this result (IN or OUT)
  `usage:` usage (in interface utilization percentage) for this top interface in this

direction during the specified time period
`inf_name:` name of this top interface

2.22 get_endpoints_name_list

Get the list of all endpoints.

- **Syntax**
  get_endpoints_name_list()

- **Synopsis**
  Returns the array of endpoint names

2.23 get_applications_name_list

Get the list of all applications.

- **Syntax**
  get_applications_name_list()

- **Synopsis**
  Returns the array of applications names

2.24 get_locations_name_list

Get the list of all locations.

- **Syntax**
  get_locations_name_list()

- **Synopsis**
  Returns the array of locations names

2.25 get_links_name_list

Get the list of all interfaces.

- **Syntax**
  get_links_name_list()

- **Synopsis**
  Returns the array of interfaces names

2.26 get_port_list

Get the list of all ports

- **Syntax**
  get_port_list()

- **Synopsis**
  Returns the array of port/protocol pairs

3. Using the Database APIs

The DBI APIs are implemented using the SOAP server on the Flow Analyze Server. The SOAP server is implemented using SOAP::Lite Perl library.

SOAP::Lite is a collection of Perl modules, which provides a simple and lightweight interface to the Simple Object Access Protocol (SOAP) on both the client and the server sides. The SOAP::Lite library allows the user to write SOAP client programs using any of the following technologies :

Perl, Apache SOAP, Apache Axis, Frontier, Microsoft SOAP, Microsoft .NET, DevelopMentor, XMethods, 4s4c, Phalanx, PocketSOAP, Kafka, SQLData, Lucin (in Java, Perl, C++, Python, VB, COM, XSLT).

The SOAP client program requires authentication to use the API. Please contact Calix for this information. SOAP server program is located at `/cgi-bin/soapserver.cgi` relative to the Tomcat web application path.

Note:
Depending on the server load, SOAP server may not give the output for the requested API. In this case, user needs to do the error checking using SOAP library function (like fault provided by SOAP::Lite).

Following is a sample client program implemented in Perl.

```
#!/usr/bin/perl -w
use SOAP::Lite;
my $client = SOAP::Lite->new();
$client->uri('urn:SoapApiBackend');
$client->proxy('https://soapuser:soapauth@10.2.1.161/cgi-
bin/soapserver.cgi',timeout  => 1800);

my $som = $client->get_endpoints_name_list();
```

```
unless ($som->fault) {
        my @intarr = $som->paramsall;
        # A simple foreach read the array
        foreach my $intval (@intarr) {
                print "topN data: $intval\n";
        }

} else {
    print join ', ',
      $som->faultcode,
      $som->faultstring,
      $som->faultdetail;
}

$som = $client->
top_applications_for_endpoint("rootswitch.example.com.","2009:06:
11T02:20:42","2009:06:11T03:20:42");
unless ($som->fault) {
        my @intarr = $som->paramsall;
        # A simple foreach read the array
        foreach my $intval (@intarr) {
                print "topN data: $intval\n";
        }

} else {
    print join ', ',
      $som->faultcode,
      $som->faultstring,
      $som->faultdetail;
}
```

To run this client program, you need to have Perl and the SOAP::Lite library installed and working on the client system. A copy of this sample program is available on the Flow Analyze Server. You can download a copy by using the **Options -> Download Files** menu, and then selecting the **Data Files** option.

Every SOAP call requires the following information.

- **address:** Address of endpoint that will handle your method call. It is specified with proxy() method using "http" protocol in our implementation

- **namespace:** Namespace (URI) of method element (specified with uri() method), which will help the SOAP server in handling your request. With our server implementation, namespace is "SoapApiBackend".

- **method and parameters:** Method name with parameters (get_endpoints_name_list() in our example).

After issuing the SOAP call, you must fetch the results by using supported library functions (such as "paramsall" in the example) and then process the results accordingly.

The following client Perl program shows the usage of one of the API calls (top_applications_for_endpoint) provided by the server. In this example, we are requesting the top applications used by the endpoint "rootswitch.example.com.", during the period from "2009:07:14 02:20:42" to "2009:07:15 02:20:42".

```
#!/usr/bin/perl -w
use SOAP::Lite;
my $client = SOAP::Lite->new();
$client->uri('urn:SoapApiBackend');
$client->proxy('https://soapuser:soapauth@10.2.1.161/cgi-
bin/soapserver.cgi',timeout  => 1800);

my $som = $client->
top_applications_for_endpoint("rootswitch.example.com.","2009:07:
14T02:20:42","2009:07:15T02:20:42");
unless ($som->fault) {
        my @intarr = $som->paramsall;
        # A simple foreach read the array
        foreach my $intval (@intarr) {
                print "topN data: $intval\n";
        }

} else {
    print join ', ',
      $som->faultcode,
      $som->faultstring,
      $som->faultdetail;
}
```

Output :

```
topN data: IN|221828842.450898|ICMP
topN data: IN|86369652.8039014|SNMP
topN data: IN|3542.40008354187|Telnet
topN data: OUT|1630441044.50776|Flow
topN data: OUT|85904184.3944339|Unmapped Ports
topN data: OUT|85722545.8540332|RADIUS
topN data: OUT|7843639.17930878|SNMP
topN data: OUT|869407.34635437|ICMP
topN data: OUT|15865.1998615265|Telnet
```

3.1 Error Handling

Dealing with error lies on the client side by checking how the server has responded.

3.1.1 Error handling in Perl based SOAP client.

With client implemented in Perl, user can use “fault” and “faultcode“ function for error checking. The faultcode() gives you information about the main reason for the error. Possible values may be

Client: you provided incorrect information in the request. This error may occur when parameters for the remote call are incorrect. Parameters may be out-of-bounds, such as negative numbers, when positive integers are expected; or of an incorrect type, for example, a string is provided where a number was expected.

Server: something is wrong on the server side. This means that provided information is correct, but the server couldn't handle the request because of temporary difficulties, for example, an unavailable database.

MustUnderstand: Header elements has mustUnderstand attribute, but wasn't understood by server. The server was able to parse the request, but the client is requesting functionality that can't be provided.

VersionMismatch: the server can't understand the version of SOAP used by the client.

3.1.2 Specific error conditions

For some specific error condition SOAP server send error as a output. These output messages are

- Invalid parameters passed to the API. Please check the user manual for API syntax.
- No Report Supported For Duration Less than 1 Hour and Greater than 12 Weeks.
- Report not available for the selected period.

4. Database Schema

The database name is 'activity’. The tables in the database activity are listed below**:**

- device_t
- direction_t
- fg_report_location_t_XXXX
- fg_report_mlink_t_XXXX
- fg_report_network_t_XXXX
- fg_report_nonid_t_XXXX
- fg_report_nonsvc_t_XXXX
- fg_report_service_t_XXXX
- fg_report_t_XXXX
- fg_report_top_ip_t_XXXX
- fg_report_top_loc_t_XXXX
- fg_report_top_mlink_t_XXXX
- fg_report_top_service_t_XXXX
- interface_t
- ip_addr_t
- location_spec_t
- location_t
- mlink_t
- network_t
- port_t
- problem_cause_t_XXXX
- problem_symptom_link_t_XXXX
- problem_t_XXXX
- profile_non_periodic_endpt_t
- profile_non_periodic_loc_t
- profile_non_periodic_mlink_t
- profile_non_periodic_network_t
- profile_non_periodic_service_t
- profile_periodic_endpt_t
- profile_periodic_loc_t
- profile_periodic_mlink_t
- profile_periodic_network_t
- profile_periodic_service_t
- prot_dom_status_t
- prot_dom_t
- protocol_t
- reason_t
- service_spec_t
- service_t
- symptom_protocol_t_XXXX

- symptom_ratio_t_XXXX
- symptom_reason_t_XXXX
- symptom_service_t_XXXX
- symptom_t_XXXX
- user_sessions_t
- user_t

In the "activity" database, there are two types of tables: **Static** tables and **Daily Generated** tables. In the above list, "XXXX" is a placeholder for a date, and the associated tables represent daily generated tables. There is a set of each such tables for each date. For example, the "fg_report_t" table for July 19, 2009 has the table name "fg_report_t_2009_7_19".

4.1 Static Tables

These are tables that are generated only once at the time of the installation of FAR. These tables store the information about various entities/objects that FAR monitors, adds, modifies, deletes, and profiles. These entities could be broadly classified into **Endpoints, Applications, Locations, Interfaces, Ports, IP addresses, and Networks**. They are stored in the database as **prot_doms, services, locations, mlinks, ports, ip_addrs, networks** respectively. There are 27 such static tables that store various information about these entities. Out of these 27, there are 10 tables that store the information about the profiles of the various objects monitored and profiled by FAR. The profiling-related tables are grouped into two types, namely **Non Periodic Profiles** and **Periodic Profiles**.

The static tables are as follows:

4.1.1 device_t
4.1.2 direction_t
4.1.3 interface_t
4.1.4 ip_addr_t
4.1.5 prot_dom_t
4.1.6 prot_dom_status_t
4.1.7 location_t
4.1.8 location_spec_t
4.1.9 mlink_t
4.1.10 network_t
4.1.11 port_t
4.1.12 protocol_t
4.1.13 reason_t
4.1.14 service_t
4.1.15 service_spec_t
4.1.16 profile_non_periodic_endpt_t
4.1.17 profile_non_periodic_loc_t
4.1.18 profile_non_periodic_mlink_t

This section provides a detailed description of all the static tables in the database.

4.1.1 device_t

Stores the information about the Flow Analyze Server.

Field	Type	Null	Key	Default	Extra
device_id	int(10) unsigned		PRI	NULL	auto_increment
device_ip	char(15)		MUL		
device_name	char(32)	YES		NULL	
device_usage	tinyint(3) unsigned			0	
device_comment	char(20)	YES		NULL	
detector_id	int(10) unsigned	YES		NULL	
traffic_feed_type	tinyint(3) unsigned			0	
sampling_rate	float			1	

- **device_id:** primary key for the device_t.
- **device_ip:** IP address of the server.
- **device_name:** name associated with the server.
- **device_comment: a**ny comment associated with the server.
- **traffic_feed_type:** type of traffic feed viz: Netflow /PCAP
- **sampling_rate:** sampling rate of the server.

4.1.2 direction_t

Stores the information about the direction of the traffic.

Field	Type	Null	Key	Default	Extra
direction_id	smallint(5) unsigned		PRI	0	
name	char(3)		UNI		

- **direction_id:** primary key for the direction_t. This direction_id is used by various other tables to specify the direction of the traffic for that specific object.
- **name:** direction in which the traffic flows viz: 0 for IN & 1 for OUT.

4.1.3 interface_t

Stores the information about interface used by the Flow Analyze Server.

Field	Type	Null	Key	Default	Extra
interface_id	int(10) unsigned		PRI	NULL	auto_increment
device_id	int(10) unsigned		MUL	0	
interface_name	char(32)				
interface_comment	char(20)	YES		NULL	

- **interface_id:** primary key for the interface_t.
- **device_id:** ID of the device to which the interface belongs to.
- **interface_name:** name of the interface viz: eth0, eth1 etc.
- **interface_comment:** any comment associated with the interface.

4.1.4 ip_addr_t

Stores the information about the discovered IP addresses on the defined network.

Field	Type	Null	Key	Default	Extra
ip_addr_id	int(10) unsigned		PRI	NULL	auto_increment
ip_addr	char(15)				
mask	smallint(5) unsigned			0	
status	smallint(5) unsigned			1	

- **ip_addr_id:** primary key for the ip_addr_t.
- **ip_addr:** IP address of the discovered node.

- **mask:** mask of the discovered IP address.
- **status:** status of the discovered IP address whether its active/deleted.viz: 1-Active and 99-Deleted.

4.1.5 prot_dom_t

Stores the information about the Endpoints in the defined network (internally referred to as prot_dom). The active IP addresses discovered in the network when mapped these are called as Endpoints.

Field	Type	Null	Key	Default	Extra
prot_dom_id	int(10) unsigned		PRI	NULL	auto_increment
prot_dom_name	varchar(32)		MUL		
prot_dom_mapped	text				
prot_dom_uid	text		MUL		
device_id	int(10) unsigned			0	
prot_dom_status_id	smallint(5) unsigned		MUL	99	

- **prot_dom_id:** primary key for the prot_dom_t.
- **prot_dom_name:** internal name of the prot_dom.
- **prot_dom_mapped:** original mapped name of the prot_dom.
- **prot_dom_uid:** display name of the prot_dom i.e. the display name of the Endpoint that shows up in the GUI.
- **device_id:** ID of the device to which the prot_dom belongs to. This is the ID from the device_t.
- **prot_dom_status_id:** status_id for the prot_dom. Whether the prot_dom is active or deleted viz: 0-BASIC, 1-ENHANCED, 99-NOT USED.
- More details from Xangati:
    1. prot_dom_mapped stored name as per mapping. prot_dom_uid is usually same as prot_dom_mapped initially but it is display name hence if user changes display name in UI, this will be reflected.
    2. In case for some reason pd does not get insert in db but data arrives, then prot_dom_status_id field in prot_dom_t will be 11, in such case prot_dom_mapped and prot_dom_uid will be blank.

4.1.6 prot_dom_status_t

Stores the information about the status mapping for the prot_dom_t.

Field	Type	Null	Key	Default	Extra
prot_dom_status_id	int(10) unsigned		PRI	0	
prot_dom_status_name	char(32)				
prot_dom_status_desc	char(20)	YES		NULL	

- **prot_dom_status_id:** primary key for prot_dom_status_t.
- **prot_dom_status_name:** status name of prot_dom viz: BASIC, ENHANCED, NOT USED
- **prot_dom_status_desc:** description for the prot_dom_status.

4.1.7 location_t

Stores the information about the Locations discovered in the defined network. A collection of Endpoints within a specified range of IP addresses is called a Location.

Field	Type	Null	Key	Default	Extra
location_id	int(10) unsigned		PRI	NULL	auto_increment
location_name	char(32)				
location_disp_name	char(32)				
status	int(10) unsigned			1	

- **location_id:** primary key for the location_t.
- **location_name**: internal name of the **Location**.
- **location_disp_name:** display name of the **Location** that appears in the GUI.
- **status:** The status of the **Location** whether it is active or deleted. viz: 1-Active and 99-Deleted.

4.1.8 location_spec_t

Stores the information about the specifications of the location subnet.

Field	Type	Null	Key	Default	Extra
location_spec_id	int(10) unsigned		PRI	NULL	auto_increment
location_id	int(10) unsigned			0	
ip_addr	char(15)				
Mask	smallint(5) unsigned			0	
Status	smallint(5) unsigned			1	

- **location_spec_id:** The Primary key for location_spec_t.
- **location_id:** ID of the location whose specification is listed.
- **ip_addr:** IP address of the location subnet.
- **mask:** mask of the location subnet.
- **status:** status of the location subnet. viz: 1-Active and 99-Deleted.

4.1.9 mlink_t

Stores the information about the Interface objects.

Field	Type	Null	Key	Default	Extra
link_id	int(10) unsigned		PRI	NULL	auto_increment
link_name	varchar(64)				
link_disp_name	varchar(64)				
link_capacity	float			0	
link_ifname	varchar(64)				
link_ifindex	varchar(64)				
link_iftype	varchar(64)				
router_ip	varchar(32)				
status	smallint(5) unsigned			1	

- **link_id:** primary key for mlink_t.
- **link_name:** internal name of the **Interface.**
- **link_disp_name:** display name of the **Interface** that shows up in GUI.
- **link_capacity:** capacity of the **Interface** in Gb.
- **link_ifname:** if_name of the **Interface.**

- **link_ifindex:** if_index of the **Interface.**
- **link_iftype:** if_type of the **Interface.**
- **router_ip:** IP address of the router to which the **Interface belongs.**
- **status:** status of **Interface** whether it is active or deleted**.** viz: 1-Active and 99-Deleted.

4.1.10 network_t

Stores the information about the Network object as per the network definition.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
network_id	int(10) unsigned		PRI	NULL	auto_increment
network_name	char(32)				
network_disp_name	char(32)				
status	smallint(5) unsigned			1	

- **network_id:** primary key for the network_t.
- **network_name:** internal name of the **Network** object.
- **network_disp_name:** display name of the **Network** object.
- **status:** status of the **Network** object whether it is active or deleted. viz: 1-Active and 99-Deleted.

4.1.11 port_t

Stores the information about the ports monitored by the FAR product.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
port_id	int(10) unsigned		PRI	NULL	auto_increment
port	int(10) unsigned			0	
protocol_id	smallint(5) unsigned			0	
status	smallint(5) unsigned			1	

- **port_id:** primary key for the port_t.
- **port:** port number of the discovered and monitored port.
- **protocol_id:** ID of the protocol associated with the port. This protocol_id is Primary key for the protocol_t.
- **status:** status of the port whether it is active or deleted. viz: 1-Active and 99-Deleted.

4.1.12 protocol_t

Stores the information about the list protocols.

Field	Type	Null	Key	Default	Extra
protocol_id	int(10) unsigned		PRI	0	
name	char(6)		UNI		

- **protocol_id:** primary key for protocol_t which is used for mapping by other tables like port_t.
- **name:** name of the protocol used. viz: IP, TCP,UDP etc.

4.1.13 reason_t

Stores the list of reasons that cause the Symptoms and Events.

Field	Type	Null	Key	Default	Extra
reason_id	int(10) unsigned		PRI	0	
reason_name	char(32)				
reason_desc	char(25)	YES		NULL	

- **reason_id:** primary key for the reason_t.
- **reason_name:** name of the reason that caused **Symptom/Event.**
- **reason_desc:** description of the reason that caused the **Symptom/Event.**

4.1.14 service_t

Stores the information about the Application object.

Field	Type	Null	Key	Default	Extra
service_id	int(10) unsigned		PRI	NULL	auto_increment
service_name	char(20)				
status	smallint(5) unsigned			1	
service_internal_name	char(20)		UNI		

- **service_id:** primary key for the service_t.
- **service_name:** internal of the **Application** object.

- **status:** status of the **Application** whether it is currently active or deleted. viz: 1-Active and 99-Deleted.
- **service_internal_name:** internal name of the **Application.**

4.1.15 service_spec_t

Stores the specification details of the Application object.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
service_spec_id	int(10) unsigned		PRI	NULL	auto_increment
protocol_id	smallint(5) unsigned		MUL	0	
port	smallint(5) unsigned			0	
service_desc	varchar(60)	YES		NULL	
service_id	int(10) unsigned			0	

- **service_spec_id:** primary key for the service_spec_t.
- **protocol_id:** ID of the protocol associated with the **Application** object.
- **port:** port number associated with the **Application** object.
- **service_desc:** description of the **Application** object.
- **service_id:** ID of the service from service_t for which the specifications are listed.

4.1.16 profile_non_periodic_endpt_t

Store the information about the Non Periodic Endpoint profiles.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
profile_non_periodic_endpt_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	
profile_end_time	int(11)			0	
profile_day_status	char(3)				
thresh_type	char(5)				
min_thresh	Float			0	
max_thresh	Float			0	
rate	Float			-1	

- **profile_non_periodic_endpt_id:** primary key for profile_non_periodic_endpt_t
- **obj_id:** ID of the **Endpoint** being profiled.
- **profile_dir:** direction of the traffic of the **Endpoint** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_day_status:** status of the day when the profiling is done. viz: ON day or OFF day.
- **thresh_type:** type of the threshold for the **Endpoint** being profiled. viz: bps, pps, nodes.
- **min_thresh:** minimum value of the threshold of the **Endpoint** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Endpoint** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** Sampling rate of the **Endpoint** at the time of profiling.

4.1.17 profile_non_periodic_loc_t

Stores the information about the Non Periodic Location profiles.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
profile_non_periodic_loc_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	
profile_end_time	int(11)			0	
profile_day_status	char(3)				
thresh_type	char(5)				
min_thresh	float			0	
max_thresh	float			0	
rate	float			-1	

- **profile_non_periodic_loc_id:** primary key for profile_non_periodic_loc_t
- **obj_id:** ID of the **Location** being profiled.
- **profile_dir:** direction of the traffic of the **Location** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_day_status:** status of the day when the profiling is done. viz: ON day or OFF day.
- **thresh_type:** type of the threshold for the **Location** being profiled. viz: bps, pps, nodes.
- **min_thresh:** minimum value of the threshold of the **Location** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Location** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** sampling rate of the **Location** at the time of profiling.

4.1.18 profile_non_periodic_mlink_t

Stores the information about the Non Periodic Interface profiles.

Field	Type	Null	Key	Default	Extra
profile_non_periodic_mlink_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	
profile_end_time	int(11)			0	
profile_day_status	char(3)				
thresh_type	char(5)				
min_thresh	Float			0	
max_thresh	Float			0	
rate	Float			-1	

- **profile_non_periodic_mlink_id:** primary key for profile_non_periodic_mlink_t
- **obj_id:** ID of the **Interface** being profiled.
- **profile_dir:** direction of the traffic of the **Interface** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_day_status:** status of the day when the profiling is done. viz: ON day or OFF day.
- **thresh_type:** type of the threshold for the **Interface** being profiled. viz: bps, pps, nodes.
- **min_thresh:** minimum value of the threshold of the **Interface** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Interface** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** sampling rate of the **Interface** at the time of profiling.

4.1.19 profile_non_periodic_network_t

Stores the information about the Non Periodic Network profiles.

Field	Type	Null	Key	Default	Extra
profile_non_periodic_network_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	
profile_end_time	int(11)			0	
profile_day_status	char(3)				
thresh_type	char(5)				
min_thresh	float			0	
max_thresh	float			0	
Rate	float			-1	

- **profile_non_periodic_network_id:** primary key for profile_non_periodic_network_t.
- **obj_id:** ID of the **Network** being profiled.
- **profile_dir:** direction of the traffic of the **Network** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_day_status:** status of the day when the profiling is done. viz: ON day or OFF day.
- **thresh_type:** type of the threshold for the **Network** being profiled. viz: bps, pps, nodes.
- **min_thresh:** minimum value of the threshold of the **Network** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Network** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** sampling rate of the **Network** at the time of profiling.

4.1.20 profile_non_periodic_service_t

Stores the information about the Non Periodic Application profiles.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
profile_non_periodic_service_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	
profile_end_time	int(11)			0	
profile_day_status	char(3)				
thresh_type	char(5)				
min_thresh	float			0	
max_thresh	float			0	
Rate	float			-1	

- **profile_non_periodic_service_id:** primary key for profile_non_periodic_service_t.
- **obj_id:** ID of the **Application** being profiled.
- **profile_dir:** direction of the traffic of the **Application** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_day_status:** status of the day when the profiling is done. viz: ON day or OFF day.
- **thresh_type:** type of the threshold for the **Application** being profiled. viz: bps, pps, nodes.
- **min_thresh:** minimum value of the threshold of the **Application** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Application** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** sampling rate of the **Application** at the time of profiling.

4.1.21 profile_periodic_endpt_t

Stores information about the Periodic Endpoint profiles.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
profile_periodic_endpt_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	
profile_end_time	int(11)			0	
profile_hour	tinyint(4)			0	
profile_hour_status	char(3)				
thresh_type	char(5)				
min_thresh	float			0	
max_thresh	float			0	
rate	float			-1	

- **profile_periodic_endpt_id:** primary key for profile_periodic_endpoint_t.
- **obj_id:** ID of the **Endpoint** being profiled.
- **profile_dir:** direction of the traffic of the **Endpoint** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_hour:** refers to the hour that the profile applies.
- **profile_hour_status:** status of the hour that the profile applies. viz: ON hour or OFF hour.
- **thresh_type:** type of the threshold for the **Endpoint** being profiled. viz: bps, pps, nodes.
- **min_thresh:** minimum value of the threshold of the **Endpoint** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Endpoint** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** sampling rate of the **Endpoint** at the time of profiling.

4.1.22 profile_periodic_loc_t

Stores the information about the Periodic Location profiles.

Field	Type	Null	Key	Default	Extra
profile_periodic_loc_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	
profile_end_time	int(11)			0	
profile_hour	tinyint(4)			0	
profile_hour_status	char(3)				
thresh_type	char(5)				
min_thresh	float			0	
max_thresh	float			0	
rate	float			-1	

- **profile_periodic_loc_id:** primary key for profile_periodic_loc_t.
- **obj_id:** ID of the **Location** being profiled.
- **profile_dir:** direction of the traffic of the **Location** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_hour:** refers to the hour that the profile applies.
- **profile_hour_status:** status of the hour that the profile applies. viz: ON hour or OFF hour.
- **thresh_type:** type of the threshold for the **Location** being profiled. viz: bps, pps,nodes.
- **min_thresh:** minimum value of the threshold of the **Location** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Location** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** sampling rate of the **Location** at the time of profiling.

4.1.23 profile_periodic_mlink_t

Stores the information about the Periodic Interface profiles.

Field	Type	Null	Key	Default	Extra
profile_periodic_mlink_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	
profile_end_time	int(11)			0	
profile_hour	tinyint(4)			0	
profile_hour_status	char(3)				
thresh_type	char(5)				
min_thresh	float			0	
max_thresh	float			0	
rate	float			-1	

- **profile_periodic_mlink_id:** primary key for profile_periodic_mlink_t.
- **obj_id:** ID of the **Interface** being profiled.
- **profile_dir:** direction of the traffic of the **Interface** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_hour:** refers to the hour that the profile applies.
- **profile_hour_status:** status of the hour that the profile applies. viz: ON hour or OFF hour.
- **thresh_type:** type of the threshold for the **Interface** being profiled. viz: bps, pps, nodes.
- **min_thresh:** minimum value of the threshold of the **Interface** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Interface** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** sampling rate of the **Interface** at the time of profiling.

4.1.24 profile_periodic_network_t

Stores the information about the Periodic Network profiles.

Field	Type	Null	Key	Default	Extra

profile_periodic_network_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	

profile_end_time
profile_hour
	int(11)
	tinyint(4)
				0
				0	

profile_hour_status	char(3)				
thresh_type	char(5)				
min_thresh	float			0	
max_thresh	float			0	
rate	float			-1	

- **profile_periodic_network_id:** primary key for profile_periodic_network_t.
- **obj_id:** ID of the **Network** being profiled.
- **profile_dir:** direction of the traffic of the **Network** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_hour:** refers to the hour that the profile applies.
- **profile_hour_status:** status of the hour that the profile applies. viz: ON hour or OFF hour.
- **thresh_type:** type of the threshold for the **Network** being profiled. viz: bps, pps, nodes.
- **min_thresh:** minimum value of the threshold of the **Network** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Network** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** sampling rate of the **Network** at the time of profiling.

4.1.25 profile_periodic_service_t

Stores the information about the Periodic Application profiles.

Field	Type	Null	Key	Default	Extra
profile_periodic_service_id	int(10) unsigned		PRI	NULL	auto_increment
obj_id	int(10) unsigned		MUL	0	
profile_dir	smallint(5) unsigned			0	
profile_start_time	int(10) unsigned			0	
profile_end_time	int(11)			0	
profile_hour	tinyint(4)			0	
profile_hour_status	char(3)				
thresh_type	char(5)				
min_thresh	float			0	
max_thresh	float			0	
rate	float			-1	

- **profile_periodic_service_id:** primary key for profile_periodic_service_t.
- **obj_id:** ID of the **Application** being profiled.
- **profile_dir:** direction of the traffic of the **Application** being profiled.
- **profile_start_time:** time at which the profiling started.
- **profile_end_time:** time at which the profiling ended, -3 indicates the current profile.
- **profile_hour:** refers to the hour that the profile applies.
- **profile_hour_status:** status of the hour that the profile applies. viz: ON hour or OFF hour.
- **thresh_type:** type of the threshold for the **Application** being profiled. viz: bps, pps, nodes.
- **min_thresh:** minimum value of the threshold of the **Application** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **max_thresh:** maximum value of the threshold of the **Application** being profiled. Its unit depends on the thresh_type. viz: bps for bps, Number of nodes for nodes, interface utilization percentage for utilization.
- **rate:** sampling rate of the **Application** at the time of profiling.

4.1.26 user_t

Stores the information about the various users of the FAR Product.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
user_id	int(10) unsigned		PRI	NULL	auto_increment
user_name	varchar(40)				
user_type	char(1)				
status	int(10) unsigned			1	

- **user_id:** primary key for the user_t to uniquely identify the user.
- **user_name:** name of the user.
- **user_type:** type of the user whether the user is **Provider** user or **Customer** user. viz: P-Provider and C-Customer.
- **status:** status of the user whether the user is active or deleted. viz: 1-Active and 99-Deleted.

4.1.27 user_sessions_t

Stores the information about the login sessions of the users.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
session_id	int(10) unsigned		PRI	NULL	auto_increment
session_ip	varchar(16)				
user_id	int(10) unsigned			0	
session_start_ts	int(10)			0	
session_end_ts	int(10)			0	

- **session_id:** primary key for the user_sessions_t.
- **session_ip:** IP address of the system from where the customer does login.
- **user_id:** ID of the user whose session details are being listed.
- **session_start_ts:** timestamp when the session started.
- **session_end_ts:** timestamp when the session ended.

4.2 Daily Generated Tables

The **Daily Generated tables** are created daily. These tables store the information about various objects extracted from the daily generated traffic. There are 38 such daily generated tables. These could be roughly grouped into three types, namely the **fg_report related** tables, the **event/symptom** related tables and the **daily summary** tables. There are 23 **fg_report related** tables, 8 **event/symptom** related tables that store the information about the **Events** and **Symptoms**, and 7 **daily summary** tables which store the daily generated information in summarized format.

In the list, "XXXX" is a placeholder for a date, and the associated tables represent daily generated tables. There is a set of each such table for each date. For example, the "fg_report_t" table for January 19, 2011 has the table name "fg_report_t_2011_1_19".

Each daily generated table is created by inheriting the property of respective parent table. This parent and child table relationship is there to implement the table merging by querying the parent table. Parent table have the same name as Daily generated tables except for the date placeholder. These parent tables are created only once at the time of fresh installation.
For example if fg_report_t_XXXX is daily generated table, then the respective parent table name is fg_report_t. This rule is applied to all the daily generated tables.

The Daily Generated tables are as follows:

4.2.1 fg_report_location_t_XXXX

Stores the information about the traffic for locations. This information is extracted from the daily generated traffic for all **locations.** A value of -3 indicates that data is not applicable for locations.

Column	**Type**	**Modifiers**
location_id	integer	not null
fg_report_time	integer	not null
rate	real	not null default 1.0
direction	smallint	not null default 0
count	smallint	not null default 0.0
bps	real	not null default 0.0
nodes	integer	not null
app_aff	integer	not null default 0
pps	real	not null default 0.0
loc_aff	integer	not null default 0
ml_aff	integer	not null default 0
ml_util	real	not null default (-1)

fg_date	date	

1. **location_id:** ID of the location from location_t table, for which the traffic details are being stored in this table.
2. **fg_report_time:** timestamp at which the traffic information is generated.
3. **rate:** sampling rate for that **Location** at time when the traffic information is generated.
4. **direction:** traffic direction for which statistics are given in that record.
5. **count:** number of seconds over which the traffic statistics for that database record apply.
6. **bps:** bandwidth consumption of the **Location** in Bytes per second. This is sampled value and thus needs the sampling rate to be applied to get the true value.
7. **nodes:** number of Endpoints related with the **Location**.
8. **app_aff:** number of Application related with the **Location**.
9. **pps:** number of packets flowing in specified direction for the **Location** in Packets per second (pps).
10. **loc_aff:** number of Location related with the **Location**.
11. **ml_aff:** number of Interfaces related with the **Location**.
12. **ml_util:** Interface utilization of the **Location** in percentage.
13. **fg_date:** date constrains added to implement parent-child relationship in PostgreSQL.

4.2.2 fg_report_mlink_t_XXXX

Stores the information about the traffic for an **Interface**. This information is extracted from the daily generated traffic for that **Interface.** A value of -3 indicates that the value is not applicable for this type of object.

Column	**Type**	**Modifiers**
link_id	integer	not null
fg_report_time	integer	not null
rate	real	not null default 1.0
direction	smallint	not null default 0
count	smallint	not null default 0.0
bps	real	not null default 0.0
nodes	integer	not null
app_aff	integer	not null default 0
pps	real	not null default 0.0
loc_aff	integer	not null default 0
ml_aff	integer	not null default 0
ml_util	real	not null default (-1)
fg_date	date	

- **link_id:** ID of the Interface (mlink) from mlink_t table, for which the traffic details are being stored in this table.
- **fg_report_time:** timestamp at which the traffic information is generated.
- **rate:** sampling rate for that **Interface** at time when the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **bps:** bandwidth consumption of the **Interface** in Bytes per second. This is sampled value and thus needs the sampling rate to be applied to get the true value.
- **nodes:** number of Endpoints related with the **Interface**.
- **app_aff:** number of Application related with the **Interface**.
- **pps:** number of packets flowing in specified direction for the **Interface** in Packets per second (pps).
- **loc_aff:** number of Location related with the **Interface**.
- **ml_aff:** number of Interfaces related with the **Interface**.
- **ml_util:** utilization of the **Interface** in interface utilization percentage.
- **fg_date:** date constrains added to implement parent-child relationship in PostgreSQL.

4.2.3 fg_report_network_t_XXXX

Stores the information about the traffic for a **Network**. This information is extracted from the daily generated traffic for that **Network**. A value of -3 indicates that the value is not applicable for this type of object.

Column	Type	Modifiers
network_id	integer	not null
fg_report_time	integer	not null
rate	real	not null default 1.0
direction	smallint	not null default 0
count	smallint	not null default 0.0
bps	real	not null default 0.0
unmap_bps	real	not null default 0.0
app_aff	integer	not null default 0
nodes	integer	not null
pps	real	not null default 0.0
loc_aff	integer	not null default 0
ml_aff	integer	not null default 0
ml_util	real	not null default (-1)
fg_date	date	

- **network_id:** ID of the **Network** from network_t table, for which the traffic details are being stored in this table.
- **fg_report_time:** timestamp at which the traffic information is generated.
- **rate:** sampling rate for that **Network** at time when the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **bps:** bandwidth consumption of the mapped traffic **Network** in Bytes per second. This is sampled value and thus needs the sampling rate to be applied to get the true value.
- **unmap_bps:** bandwidth consumption of the unmapped traffic in **Network** in Bytes per second. This is sampled value and thus needs the sampling rate to be applied to get the true value.
- **app_aff:** number of Application related with the **Network**.
- **nodes:** number of Endpoints related with the **Network**.
- **pps:** number of packets flowing in specified direction for the **Network** in packets per second (pps).
- **loc_aff:** number of Location related with the **Network**.
- **ml_aff:** number of Interfaces related with the **Network**.
- **ml_util:** utilization of the **Network** in interface utilization percentage.
- **fg_date:** date constrains added to implement parent-child relationship in PostgreSQL.

4.2.4 fg_report_nonid_t_XXXX

Stores the information about the traffic for an **IP address** from ip_addr_t which has not yet been mapped into an Endpoint. This information is extracted from the daily generated traffic for that **IP address.** A value of -3 indicates that the value is not applicable for this type of object.

Column	**Type**	**Modifiers**
obj_id	integer	not null
fg_report_time	integer	not null
rate	real	not null default 1.0
direction	smallint	not null default 0
count	smallint	not null default 0.0
bps	real	not null default 0.0
app_aff	integer	not null default 0
nodes	integer	not null
pps	real	not null default 0.0
loc_aff	integer	not null default 0.0
ml_aff	integer	not null default 0.0

ml_util	real	not null default (-1)
fg_date	date	

- **obj_id:** ID of the object from ip_addr_t table, for which the traffic details are being stored in this table.
- **fg_report_time:** timestamp at which the traffic information is generated.
- **rate:** sampling rate for that IP at time when the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record. .
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **bps:** bandwidth consumption of the **unmapped IP** in Bytes per second. This is sampled value and thus needs the sampling rate to be applied to get the true value.
- **app_aff:** number of Application related with the **unmapped IP**.
- **nodes:** number of Endpoints related with the **unmapped IP**.
- **pps:** number of packets flowing in specified direction for the **unmapped IP** in Packets per second (pps).
- **loc_aff:** number of Location related with the **unmapped IP**.
- **ml_aff:** number of Interfaces related with the **unmapped IP**.
- **ml_util:** utilization of the **unmapped IP** in interface utilization percentage.
- **fg_date:** date constrains added to implement parent-child relationship in PostgreSQL.

4.2.5 fg_report_nonsvc_t_XXXX

Stores the information about the traffic for a **Port** object. This information is extracted from the daily generated traffic for that **Port.** A value of -3 indicates that the value is not applicable for this type of object.

Column	Type	Modifiers
obj_id	integer	not null
fg_report_time	integer	not null
rate	real	not null default 1.0
direction	smallint	not null default 0
count	smallint	not null default 0.0
bps	real	not null default 0.0
nodes	integer	not null default 0
app_aff	integer	not null default 0
pps	real	not null default 0.0
loc_aff	integer	not null default 0
ml_aff	integer	not null default 0
ml_util	real	not null default (-1)
fg_date	date	

- **obj_id:** ID of the port from port_t table, for which the traffic details are being stored in this table.
- **fg_report_time:** timestamp at which the traffic information is generated.
- **rate:** sampling rate for that **Port** at time when the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **bps:** bandwidth consumption of the **Port** in Bytes per second. This is sampled value and thus needs the sampling rate to be applied to get the true value.
- **nodes:** number of Endpoints related with the **Port**.
- **app_aff:** number of Application related with the **Port**.
- **pps:** number of packets flowing in specified direction for the **Port** in packets per second (pps).
- **loc_aff:** number of Location related with the **Port**.
- **ml_aff:** number of Interfaces related with the **Port**.
- **ml_util:** utilization of the **Port** in interface utilization percentage.
- **fg_date:** date constrains added to implement parent-child relationship in PostgreSQL.

4.2.6 fg_report_service_t_XXXX

Stores the information about the traffic for an **Application** (service). This information is extracted from the daily generated traffic for that **Application.** A value of -3 indicates that the value is not applicable for this type of object.

Column	Type	Modifiers
obj_id	integer	not null
fg_report_time	integer	not null
rate	real	not null default 1.0
direction	smallint	not null default 0
count	smallint	not null default 0.0
bps	real	not null default 0.0
nodes	integer	not null default 0
app_aff	integer	not null default 0
pps	real	not null default 0.0
loc_aff	integer	not null default 0
ml_aff	integer	not null default 0
ml_util	real	not null default (-1)
fg_date	date	

- **obj_id:** ID of the service from service_t table, for which the traffic details are being stored in this table.

- **fg_report_time:** timestamp at which the traffic information is generated.
- **rate:** sampling rate at time when the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **bps:** bandwidth consumption of the **Application** in Bytes per second. This is sampled value and thus needs the sampling rate to be applied to get the true value.
- **nodes:** number of Endpoints related with the **Application**.
- **app_aff:** number of Application related with the **Application**.
- **pps:** number of packets flowing in specified direction for the **Application** in Packets per second (pps).
- **loc_aff:** number of Location related with the **Application**.
- **ml_aff:** number of Interfaces related with the **Application**.
- **ml_util:** utilization of the **Application** in interface utilization percentage.
- **fg_date:** date constrains added to implement parent-child relationship in PostgreSQL.

4.2.7 fg_report_t_XXXX

Stores the information about the traffic for an **Endpoint** (prot_dom). This information is extracted from the daily generated traffic for that **Endpoint.** A value of -3 indicates that the value is not applicable for this type of object.

Column	**Type**	**Modifiers**
prot_dom_id	integer	not null
fg_report_time	integer	not null
rate	real	not null default 1.0
direction	smallint	not null default 0
count	smallint	not null default 0.0
bps	real	not null default 0.0
app_aff	integer	not null default 0.0
nodes	integer	not null
pps	real	not null default 0.0
loc_aff	integer	not null
ml_aff	integer	not null
ml_util	real	not null default (-1)
fg_date	date	

- **prot_dom_id:** ID of the prot_dom from prot_dom_t table for which the traffic details are being stored.
- **fg_report_time:** timestamp at which the traffic information is generated.

- **rate:** sampling rate for **Endpoint** at time when the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **bps:** bandwidth consumption of the **Endpoint** Bytes per second. This is sampled value and thus needs the sampling rate to be applied to get the true value.
- **app_aff:** number of Application related with the **Endpoint**.
- **nodes:** number of Endpoints related with the **Endpoint**.
- **pps:** number of packets flowing in specified direction for the **Endpoint** in packets per second (pps).
- **loc_aff:** number of Location related with the **Endpoint**.
- **ml_aff:** number of Interfaces related with the **Endpoint** .
- **ml_util:** utilization of the **Endpoint** in interface utilization percentage.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.8 fg_report_loc_top_ip_t_XXXX

Stores the information about **TOP IPs/Identities** associated with discovered groups in the configured network. This information is extracted from the daily generated traffic for the discovered groups.

Column	**Type**	**Modifiers**
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
top_id	integer[ ]	
traffic_rate	real[ ]	
top_id_type	character(1)[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP IP** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **top_id:** numeric ids referring to primary key in prot_dom_t table or ip_addr_t table. This list of top ids is stored in array format and is associated with id type as indicated by field 'top_id_type' which also store the id types in array format.

- **traffic_rate:** value for bandwidth usage in array format. Stores bandwidth usage associated with each top IP/Identity listed in 'top_id' field.
- **top_id_type:** storing letters 'I' or 'E' in array format to indicate each object listed by top_id field is IP address or identity.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.9 fg_report_loc_top_loc_t_XXXX

Stores the information about **TOP Groups** associated with discovered groups in the configured network. This information is extracted from the daily generated traffic for the discovered groups.

Column	Type	Modifiers
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
location_id	integer[ ]	
traffic_rate	real[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP Group** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **location_id:** numeric ids referring to primary key in location_t table. This list of group ids is stored in array format and is associated with values stored in 'traffic_rate' field.
- **traffic_rate:** value for bandwidth usage in array format. Stores bandwidth usage associated with each top group listed in 'location_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.10 fg_report_loc_top_mlink_t_XXXX

Stores the information about **TOP Interfaces** associated with discovered groups in the configured network. This information is extracted from the daily generated traffic for the discovered groups.

Column	Type	Modifiers
obj_id	integer	not null

report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
link_id	integer[ ]	
traffic_rate	real[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP Interfaces** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **link_id:** numeric ids referring to primary key in mlink_t table. This list of interface ids is stored in array format and is associated with values stored in 'traffic_rate' field.
- **traffic_rate:** value for link utilization in array format. Stores link utilization associated with each top interface listed in 'link_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.11 fg_report_loc_top_service_t_XXXX

Stores the information about **TOP Applications** associated with discovered groups in the configured network. This information is extracted from the daily generated traffic for the discovered groups.

Column	**Type**	**Modifiers**
abnormal_obj_id	integer	not null
fg_report_time	integer	not null
direction	character(1)	not null
count	smallint	not null default 0.0
service_id	integer[ ]	
bps	real[ ]	
fg_date	date	not null

- **abnormal_obj_id:** ID of the discovered object for which the TOP Application traffic details are being stored in this table.
- **fg_report_time:** timestamp at which the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **service_id:** numeric ids referring to primary key in service_t table. This list of

ids is stored in array format and is associated with the value stored in 'bps' field.

- **bps:** value for bandwidth usage in array format. Stores bandwidth usage associated with each top application listed in 'service_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.12 fg_report_ml_top_ip_t_XXXX

Stores the information about **TOP IPs/Identities** associated with discovered interfaces in the configured network. This information is extracted from the daily generated traffic for the discovered interfaces.

Column	Type	Modifiers
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
top_id	integer[ ]	
traffic_rate	real[ ]	
top_id_type	character(1)[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP IP** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **top_id:** numeric ids referring to primary key in prot_dom_t table or ip_addr_t table. This list of top ids is stored in array format and is associated with id type as indicated by field 'top_id_type' which also store the id types in array format.
- **traffic_rate:** value for link utilization in array format. Stores link utilization associated with each top IP/Identity listed in 'top_id' field.
- **top_id_type:** storing letters 'I' or 'E' in array format to indicate each object listed by top_id field is IP address or identity.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.13 fg_report_ml_top_loc_t_XXXX

Stores the information about **TOP Groups** associated with discovered interfaces in the configured network. This information is extracted from the daily generated traffic for the discovered interfaces.

Column	Type	Modifiers
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
location_id	integer[ ]	
traffic_rate	real[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP Group** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **location_id:** numeric ids referring to primary key in location_t table. This list of group ids is stored in array format and is associated with values stored in 'traffic_rate' field.
- **traffic_rate:** value for link utilization in array format. Stores link utilization associated with each top group listed in 'location_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.14 fg_report_ml_top_mlink_t_XXXX

Stores the information about **TOP Interfaces** associated with discovered interfaces in the configured network. This information is extracted from the daily generated traffic for the discovered interfaces.

Column	Type	Modifiers
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
link_id	integer[ ]	
traffic_rate	real[ ]	
fg_date	date	

- **obj_id:** ID of the discovered object for which the **TOP Interfaces** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.

- **dir:** traffic direction for which statistics are given in that record.
- **link_id:** numeric ids referring to primary key in mlink_t table. This list of interface ids is stored in array format and is associated with values stored in 'traffic_rate' field.
- **traffic_rate:** value for link utilization in array format. Stores link utilization associated with each top interface listed in 'link_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.15 fg_report_ml_top_service_t_XXXX

Stores the information about **TOP Applications** associated with discovered interfaces in the configured network. This information is extracted from the daily generated traffic for the discovered interfaces.

Column	Type	Modifiers
abnormal_obj_id	integer	not null
fg_report_time	integer	not null
direction	character(1)	not null
count	smallint	not null default 0.0
service_id	integer[ ]	
bps	real[ ]	
fg_date	date	not null

- **abnormal_obj_id:** ID of the discovered object for which the TOP Application traffic details are being stored in this table.
- **fg_report_time:** timestamp at which the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **service_id:** numeric ids referring to primary key in service_t table. This list of ids is stored in array format and is associated with the value stored in 'bps' field.
- **bps:** value for link utilization in array format. Stores link utilization associated with each top application listed in 'service_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.16 fg_report_pd_top_ip_t_XXXX

Stores the information about **TOP IPs/Identities** associated with discovered identities in the configured network. This information is extracted from the daily generated traffic for the discovered identities.

Column	Type	Modifiers

obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
top_id	integer[ ]	
traffic_rate	real[ ]	
top_id_type	character(1)[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP IP** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **top_id:** numeric ids referring to primary key in prot_dom_t table or ip_addr_t table. This list of top ids is stored in array format and is associated with id type as indicated by field 'top_id_type' which also store the id types in array format.
- **traffic_rate:** value for bandwidth usage in array format. Stores bandwidth usage associated with each top IP/Identity listed in 'top_id' field.
- **top_id_type:** storing letters 'I' or 'E' in array format to indicate each object listed by top_id field is IP address or identity.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.17 fg_report_pd_top_mlink_t_XXXX

Stores the information about **TOP Interfaces** associated with discovered identities in the configured network. This information is extracted from the daily generated traffic for the discovered identities.

Column	**Type**	**Modifiers**
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
link_id	integer[ ]	
traffic_rate	real[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP Interfaces** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.

- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **link_id:** numeric ids referring to primary key in mlink_t table. This list of interface ids is stored in array format and is associated with values stored in 'traffic_rate' field.
- **traffic_rate:** value for link utilization in array format. Stores link utilization associated with each top interface listed in 'link_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.18 fg_report_pd_top_service_t_XXXX

Stores the information about **TOP Applications** associated with discovered identities in the configured network. This information is extracted from the daily generated traffic for the discovered identities.

Column	**Type**	**Modifiers**
abnormal_obj_id	integer	not null
fg_report_time	integer	not null
direction	character(1)	not null
count	smallint	not null default 0.0
service_id	integer[ ]	
bps	real[ ]	
fg_date	date	not null

- **abnormal_obj_id:** ID of the discovered object for which the TOP Application traffic details are being stored in this table.
- **fg_report_time:** timestamp at which the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **service_id:** numeric ids referring to primary key in service_t table. This list of ids is stored in array format and is associated with the value stored in 'bps' field.
- **bps:** value for bandwidth usage in array format. Stores bandwidth usage associated with each top application listed in 'service_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.19 fg_report_port_top_ip_t_XXXX

Stores the information about **TOP IPs/Identities** associated with discovered ports in the configured network. This information is extracted from the daily generated traffic for the discovered ports.

Column	Type	Modifiers
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
top_id	integer[ ]	
traffic_rate	real[ ]	
top_id_type	character(1)[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP IP** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **top_id:** numeric ids referring to primary key in prot_dom_t table or ip_addr_t table. This list of top ids is stored in array format and is associated with id type as indicated by field 'top_id_type' which also store the id types in array format.
- **traffic_rate:** value for bandwidth usage in array format. Stores bandwidth usage associated with each top IP/Identity listed in 'top_id' field.
- **top_id_type:** storing letters 'I' or 'E' in array format to indicate each object listed by top_id field is IP address or identity.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.20 fg_report_rt_top_service_t_XXXX

Stores the information about **TOP Applications** across the configured network. This information is extracted from the daily generated traffic in the configured network.

Column	Type	Modifiers
abnormal_obj_id	integer	not null
fg_report_time	integer	not null
direction	character(1)	not null
count	smallint	not null default 0.0
service_id	integer[ ]	
bps	real[ ]	
fg_date	date	not null

- **abnormal_obj_id:** ID of the discovered object for which the **TOP Application** traffic details are being stored in this table.

- **fg_report_time:** timestamp at which the traffic information is generated.
- **direction:** traffic direction for which statistics are given in that record.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **service_id:** numeric ids referring to primary key in service_t table. This list of ids is stored in array format and is associated with the value stored in 'bps' field.
- **bps:** value for bandwidth usage in array format. Stores bandwidth usage associated with each top application listed in 'service_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.21 fg_report_svc_top_ip_t_XXXX

Stores the information about **TOP IPs/Identities** associated with discovered applications in the configured network. This information is extracted from the daily generated traffic for the discovered applications.

Column	**Type**	**Modifiers**
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
top_id	integer[ ]	
traffic_rate	real[ ]	
top_id_type	character(1)[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP IP** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **top_id:** numeric ids referring to primary key in prot_dom_t table or ip_addr_t table. This list of top ids is stored in array format and is associated with id type as indicated by field 'top_id_type' which also store the id types in array format.
- **traffic_rate:** value for bandwidth usage in array format. Stores bandwidth usage associated with each top IP/Identity listed in 'top_id' field.
- **top_id_type:** storing letters 'I' or 'E' in array format to indicate each object listed by top_id field is IP address or identity.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.22 fg_report_svc_top_loc_t_XXXX

Stores the information about **TOP Groups** associated with discovered applications in the configured network. This information is extracted from the daily generated traffic for the discovered applications.

Column	**Type**	**Modifiers**
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
location_id	integer[ ]	
traffic_rate	real[ ]	
fg_date	date	not null

- **obj_id:** ID of the discovered object for which the **TOP Group** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **location_id:** numeric ids referring to primary key in location_t table. This list of group ids is stored in array format and is associated with values stored in 'traffic_rate' field.
- **traffic_rate:** value for bandwidth usage in array format. Stores bandwidth usage associated with each top group listed in 'location_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.23 fg_report_svc_top_mlink_t_XXXX

Stores the information about **TOP Interfaces** associated with discovered applications in the configured network. This information is extracted from the daily generated traffic for the discovered applications.

Column	**Type**	**Modifiers**
obj_id	integer	not null
report_time	integer	not null
count	smallint	not null
dir	character(1)	not null
link_id	integer[ ]	
traffic_rate	real[ ]	
fg_date	date	

- **obj_id:** ID of the discovered object for which the **TOP Interfaces** traffic details are being stored in this table.
- **report_time:** timestamp at which the traffic information is generated.
- **count:** number of seconds over which the traffic statistics for that database record apply.
- **dir:** traffic direction for which statistics are given in that record.
- **link_id:** numeric ids referring to primary key in mlink_t table. This list of interface ids is stored in array format and is associated with values stored in 'traffic_rate' field.
- **traffic_rate:** value for link utilization in array format. Stores link utilization associated with each top interface listed in 'link_id' field.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.24 symptom_t_XXXX

Stores the **Symptoms** generated by the discovered objects which eventually generate the **Events.**

Column	**Type**	**Modifiers**
symptom_id	integer	not null
table_no	integer	not null
abnormal_obj_type	character varying(5)	not null
abnormal_obj_id	integer	not null
dir	smallint	not null default 0
attack_id	integer	not null
start_detect	integer	not null
most_recent_detect	integer	not null
low_bps_thresh	real	
low_pps_thresh	real	
low_nodes_thresh	real	
high_bps_thresh	real	
high_pps_thresh	real	
high_nodes_thresh	real	
bps_hyperness	real	
pps_hyperness	real	
nodes_hyperness	real	
bps_hyponess	real	
pps_hyponess	real	
nodes_hyponess	real	
severity	real	default 0
min_severity	real	default 0

max_severity	real	default 0
status	smallint	not null default 0
fg_date	date	

- **symptom_id:** primary key for symptom_t_XXXX.
- **table_no:** table number formed depending upon the current date forms the primary key along with the **symptom_id**.
- **abnormal_obj_type**: type of the discovered object for which the symptom details are being stored.
- **abnormal_obj_id:** ID of the discovered object for which the symptom details are being stored in this table.
- **dir:** direction of traffic in which the symptom is observed for the discovered object viz **IN** and **OUT.**
- **attack_id:** attack ID for the symptom .
- **start_detect:** timestamp when the symptom started.
- **most_recent_detect:** timestamp when the symptom was most recently detected/observed.
- **low_bps_thresh:** low threshold limit of bps for an object, below which the symptom is generated for that object.
- **low_pps_thresh:** low threshold limit of pps for an object, below which the symptom is generated for that object.
- **low_nodes_thresh:** low threshold limit of number of nodes associated with an object, below which the symptom is generated for that object.
- **high_bps_thresh:** high threshold limit of bps for an object, above which the symptom is generated for that object.
- **high_pps_thresh:** high threshold limit of pps for an object, above which the symptom is generated for that object.
- **high_nodes_thresh:** high threshold limit of number of nodes associated with an object, above which the symptom is generated for that object.
- **bps_hyperness:** ratio of observed high bps to the defined threshold **high_bps_thresh**.
- **pps_hyperness:** ratio of observed high pps(packets) to the defined threshold **high_pps_thresh**.
- **nodes_hyperness:** ratio of observed high number of nodes to the defined threshold **high_nodes_thresh**.
- **bps_hyponess:** ratio of observed low bps to the defined threshold **low_bps_thresh.**
- **pps_hyponess:** ratio of observed low pps(packets) to the defined threshold **low_pps_thresh** limit.
- **nodes_hyponess:** ratio of observed low number of nodes to the defined **low_nodes_thresh** limit.
- **severity:** severity of the Symptom.
- **min_severity:** minimum severity of the Symptom.

- **max_severity:** maximum severity of the Symptom.
- **status:** status of the symptom whether it's still Active or Expired. 1-Active and 99-Expired.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.25 symptom_protocol_t_XXXX

Stores the information about the protocol details associated with the **Symptom.**

Column	Type	Modifiers
symptom_protocol_id	integer	not null
symptom_table_no	integer	not null
abnormal_obj_type	character varying(5)	not null
abnormal_obj_id	integer	not null
symptom_id	integer	not null
protocol_id	integer	not null
fg_date	date	

- **symptom_protocol_id:** primary key for the symptom_protocol_t_XXXX.
- **symptom_table_no:** symptom table number from the symptom_t_XXXX.
- **abnormal_obj_type:** object type for which the symptom protocol details are being stored.
- **abnormal_obj_id:** ID of the object for which the symptom protocol details are being stored in this table.
- **symptom_id:** ID of the Symptom from symptom_t_XXXX for which the symptom protocol details are being stored.
- **protocol_id:** ID of the protocol from protocol_t for which the symptom protocol details are being stored.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.26 symptom_ratio_t_XXXX

Stores the various ratio information's about the Symptom.

Column	Type	Modifiers
symptom_ratio_id	integer	not null
symptom_id	integer	not null
symptom_table_no	integer	not null
obj_type	character varying(5)	not null
obj_id	integer	not null
ratio	real	not null default 0
ratio_type	character varying(4)	not null

fg_date	date	

- **symptom_ratio_id:** primary key for symptom_ratio_t_XXXX for which the symptom ratio details are being stored.
- **symptom_id:** ID of the Symptom from symptom_t_XXXX for which the symptom ratio details are being stored.
- **symptom_table_no:** symptom table number from the symptom_t_XXXX.
- **obj_type:** object type for which the symptom ratio details are being stored.
- **obj_id:** ID of the object for which the symptom ratio details are being stored in this table.
- **ratio:** ratio of **hyperness of a measure to high threshold of the measure** or the ratio of **hyponess of a measure to the low threshold of the measure**.
- **ratio_type:** ratio type indicates the type of ratio and the object for which it is stored.

  For Example:
  '1A' indicates that the ratio is for Application Hyperness.
  '0A' indicates that the ratio is for Application Hyponess.
  '1L' indicates that the ratio is for Location Hyperness.
  '0L' indicates that the ratio is for Location Hyponess.

  Here :
  '1' stands for Hyperness and **'0'** stands for Hyponess.
  'A' indicates the object type as **Application.**
  'L' indicates the object type as **Location.**
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.27 symptom_reason_t_XXXX

Stores the information about the reason details associated with the **Symptom.**

Column	Type	Modifiers
symptom_reason_id	integer	not null
symptom_table_no	integer	not null
abnormal_obj_type	character varying(5)	not null
abnormal_obj_id	integer	not null
symptom_id	integer	not null
reason_id	integer	not null
fg_date	date	

- **symptom_reason_id:** primary key for the symptom_reason_t_XXXX.
- **symptom_table_no:** symptom table number from the symptom_t_XXXX.

- **abnormal_obj_type:** object type for which the symptom's reason details are being stored.
- **abnormal_obj_id:** ID of the object for which the symptom's reason details are being stored in this table.
- **symptom_id:** ID of the Symptom from symptom_t_XXXX for which the symptom's reason details are being stored.
- **reason_id:** ID of the reason from reason_t for which the symptom's reason details are being stored.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.28 symptom_service_t_XXXX

Stores the information about the services and ports associated with the **Symptom.**

Column	**Type**	**Modifiers**
symptom_service_id	integer	not null
symptom_table_no	integer	not null
abnormal_obj_type	character varying(5)	not null
abnormal_obj_id	integer	not null
symptom_id	integer	not null
service_port	character varying(20)	not null
fg_date	date	

- **symptom_service_id:** primary key for the symptom_service_t_XXXX.
- **symptom_table_no:** symptom table number from the symptom_t_XXXX.
- **abnormal_obj_type:** object type for which the symptom's service/port details are being stored.
- **abnormal_obj_id:** ID of the object for which the symptom's service/port details are being stored in this table.
- **symptom_id:** ID of the Symptom from symptom_t_XXXX for which the symptom's service/port details are being stored.
- **service_port:** service name or the port number for which the symptom's service/port details are being stored.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.29 problem_cause_t_XXXX

Stores the cause associated with the **Event**.

Column	**Type**	**Modifiers**
problem_cause_id	integer	not null
problem_id	bigint	not null

problem_cause	character varying(32)	not null
fg_date	date	

- **problem_cause_id:** primary key for problem_cause_t_XXXX.
- **problem_id:** ID of the Event from the problem_t table, for which the cause is being stored in this table.
- **problem_cause:** cause of the Event.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.30 problem_symptom_link_t_XXXX

Stores the association/mapping of a **Symptom** with an **Event.**

Column	Type	Modifiers
problem_symptom_link_id	integer	not null
problem_id	bigint	not null
abnormal_obj_type	character varying(5)	not null
abnormal_obj_id	integer	not null
attack_id	integer	not null
fg_date	date	

- **problem_symptom_link_id:** primary key for the problem_symptom_link_t_XXXX.
- **problem_id:** ID of the Event from problem_t_XXXX table.
- **abnormal_obj_type:** type of the object for which the Event has occurred.
- **abnormal_obj_id:** ID of the object for which the Event has occurred.
- **attack_id:** ID of the Symptom from symptom_t which is associated with the Event.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.31 problem_t_XXXX

Stores the information about the Events detected for any discovered object on the defined network.

Column	Type	Modifiers
problem_id	bigint	not null
start_detect	integer	not null
most_recent_detect	integer	not null
aspect_type	character varying(16)	
aspect_instance	character varying(64)	

aspect_likelihood	real	not null
severity	real	not null
min_severity	real	not null
max_severity	real	not null
status	smallint	not null
fg_date	date	

1. **problem_id:** primary key for the problem_t that uniquely identifies the event.
2. **start_detect:** timestamp when the Event start is detected.
3. **most_recent_detect:** timestamp when the Event was most recently detected.
4. **aspect_type:** aspect type of the Event viz SEV,ORIGIN,TARGET etc.
5. **aspect_instance:** particular instance of the specific aspect_type of the Event.
6. **aspect_likelihood:** likelihood of the aspect type for this event.
7. **severity:** severity of the Event.
8. **min_severity:** minimum severity of the Event.
9. **max_severity:** maximum severity of the Event.
10. **status:** status of the Event whether it's Active or Expired. viz: 1-Active and 99-Expired.
11. **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.32 daily_app_summary_t_XXXX

Stores the information about the daily traffic that is generated by the Applications, in a summarized format for more efficient searches.

Column	**Type**	**Modifiers**
obj_id	integer	not null
direction	smallint	not null default 0
bps_avg	real	not null default 0.0
id_aff	real	not null default 0.0
fg_date	date	

- **obj_id:** ID of the object for which the Application related traffic summary of that day is being stored.
- **direction:** direction of the traffic viz 0 for IN and 1 for OUT.
- **bps_avg:** average of the bandwidth usage for the Application in Bytes per second.
- **id_aff:** number of related Endpoints.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.33 daily_id_summary_t_XXXX

Stores the information about the daily traffic that is generated by the Endpoints, in a summarized format for more efficient searches.

Column	Type	Modifiers
obj_id	integer	not null
direction	smallint	not null default 0
bps_avg	real	not null default 0.0
app_aff	real	not null default 0.0
id_aff	real	not null default 0.0
fg_date	date	

- **obj_id:** ID of the object for which the Endpoint related traffic summary of that day is being stored.
- **direction:** direction of the traffic viz 0 for IN and 1 for OUT.
- **bps_avg:** average of the bandwidth usage for the Endpoint in Bytes per second .
- **app_aff:** number of related Application.
- **id_aff:** number of related Endpoints.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.34 daily_ip_summary_t_XXXX

Stores the information about the daily traffic that is generated by the unmapped IPs, in a summarized format for more efficient searches.

Column	Type	Modifiers
obj_id	integer	not null
direction	smallint	not null default 0
bps_avg	real	not null default 0.0
app_aff	real	not null default 0.0
fg_date	date	

- **obj_id:** ID of the object for which the IP related traffic summary of that day is being stored.
- **direction:** direction of the traffic viz 0 for IN and 1 for OUT.
- **bps_avg:** average of the bandwidth usage for the IP in Bytes per second.
- **app_aff:** number of related Application.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.35 daily_loc_summary_t_XXXX

Stores the information about the daily traffic that is generated by the Locations, in a summarized format for more efficient searches.

Column	Type	Modifiers
obj_id	integer	not null
direction	smallint	not null default 0
bps_avg	real	not null default 0.0
fg_date	date	

1. **obj_id:** ID of the object for which the Location related traffic summary of that day is being stored.
2. **direction:** direction of the traffic viz 0 for IN and 1 for OUT.
3. **bps_avg:** average of the bandwidth usage for the Location in Bytes per second.
4. **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.36 daily_mlink_summary_t_XXXX

Stores the information about the daily traffic that is generated by the Interfaces, in a summarized format for more efficient searches.

Column	Type	Modifiers
obj_id	integer	not null
direction	smallint	not null default 0
bps_avg	real	not null default 0.0
link_util	real	not null default 0.0
fg_date	date	

- **obj_id:** ID of the object for which the Interface related traffic summary of that day is being stored.
- **direction:** direction of the traffic viz 0 for IN and 1 for OUT.
- **bps_avg:** average of the bandwidth usage for the Interface in Bytes per second
- **link_util:** interface utilization percentage for the object.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.37 daily_network_summary_t_XXXX

Stores the information about the daily traffic that is generated by the Interfaces, in a summarized format for more efficient searches.

Column	Type	Modifiers
obj_id	integer	not null
direction	smallint	not null default 0
bps	real	not null default 0.0
unmap_bps	real	not null default 0.0
fg_date	date	

1. **obj_id:** ID of the object for which the Network related traffic summary of that day is being stored.
2. **direction:** direction of the traffic viz 0 for IN and 1 for OUT.
3. **bps:** average of the bandwidth usage for mapped objects in network in Bytes per second.
4. **unmap_bps:** average of the bandwidth usage for the unmapped objects in network in Bytes per second.
5. **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

4.2.38 daily_port_summary_t_XXXX

Stores the information about the daily traffic that is generated by the Ports, in a summarized format for more efficient searches.

Column	**Type**	**Modifiers**
obj_id	integer	not null
direction	smallint	not null default 0
bps_avg	real	not null default 0.0
id_aff	real	not null default 0.0
fg_date	date	

- **obj_id:** The ID of the object for which the Port related traffic summary of that day is being stored.
- **direction:** direction of the traffic viz 0 for IN and 1 for OUT.
- **bps_avg:** average of the bandwidth usage for the Port in Bytes per second.
- **id_aff:** number of related Endpoints.
- **fg_date:** date constrain added to implement parent-child relationship in PostgreSQL.

Note: The tables device_link_t and fg_report_port_top_mlink_t are part of the database but are not used, so these are not documented. In subsequent release, these tables will be deleted.

5. Exporting and Restoring the Database for Offline Data Mining

This describes how the database can be exported to a remote system (assumed to be Linux-based) and restored on the remote system for offline data mining.

5.1 Exporting the Database

Log in to the Provider UI of the server. Click on **Options -> Export Report Database** to bring up a window for exporting the database to a remote system.

1. **IP:** Type the IP address of the remote system.

2. **Port:** Type the port number on which the SSH server of the remote system is running. The default port is 22.

3. **Username:** Type the Username through which the software will connect to the remote system.

4. **Password:** Type the password for gaining access to the remote system.

5. **Directory name on Remote System:** Type the folder name in the remote system where you would like to store the database.
   Please ensure that the remote system has sufficient file space to hold the exported database. A full 12-week backup will require approximately 200 to 300 GB of disk space.

6. **Status Message:** Displays a relevant message based on the success or failure of the export. This message will be displayed only after you have clicked the **OK** button.

Click **OK** to start exporting the database or **Cancel** if you want to cancel the export request.
If the export was successful a message **Success** is displayed otherwise a **Failure** message is displayed.

Note:

- The **Directory name on Remote System** that you specify follows standard Linux file naming convention. If you specify a relative pathname, it will be relative to the user’s home directory. If you specify an absolute pathname (those that start with a leading slash character “/”), then the pathname will be relative to the root of the file system.

5.2 Restoring the Database

Upon successful export, dump files are created on the specified directory on the remote system. These files have names such as db_dump_1, db_dump_2, and so on. Use the following instructions to restore the database.

- Ensure that mysql is installed and working on the machine.

- The syntax to restore the dump files is

```
mysql -u 'username' -p'password' 'databasename' <'path of
dumpfile/dump file name'
```

Where
`username` is the user name for mysql.
`password` is the password for mysql
`databasename` is the name of the database that should exist.
`path of dumpfile/dump file name` is the path where the dump files are present.

Example:

```
mysql -u abc -ptt123 db_name < /home/calix/db_dump_1
mysql -u abc -ptt123 db_name < /home/calix/db_dump_2
```

Note:

- The above command has to be repeated for each individual dump file. Therefore, if you have five dump files, then execute the command five times, once for each file.

6. Usage Interfaces

A programmatic interface that allows an external system to retrieve usage information for endpoints. This information can be used for general usage information as well as billing based on usage.

6.1 getUse

This synchronous call is used to extract and return an array of IPDRX records. IPDRX record is an extension of IPDR record. This call enhances getIPDR call by adding additional input parameters as follows.

- Input
    - Entity Type
        - Type: String
        - The type of an entity. There are four types of entities in FA, which are 'endpoint', 'application', 'interface' and 'location'. Therefore, users can only provide the call with these four types of string.
        - This parameter is mandatory.
    - Entity ID
        - Type: String
        - The identification of an entity. This is the equivalence of the Endpoint parameter in getIPDR call if entity type is 'endpoint'.
        - Omit parameter to return ALL entities with a certain entity type found.
    - Start Time
        - Type: Date
        - The UTC time from which usage information is requested. Actual returned time may be earlier as it starts at the boundary of the current interval. The start time is inclusive.
    - End Time
        - Type: Date
        - The UTC time up to which usage information is requested. The end time is exclusive.
    - Interval
        - Type: String
        - The valid values are 'daily', 'hourly' and 'monthly'.
    - Dimension
        - Type: String
        - The dimension of result data set. This parameter is used to separate the result data set in accordance with different entity types. If the user

        provisions ‘application’ as a dimension, the result data set will be separated by different applications running on the entity.
    - Mapping Detail
        - Type: Boolean
        - Provide this flag if you want the mapping method returned with each endpoint name. This allows a script to interpret and parse the endpoint name depending on how it is obtained.

- Output
    - This function returns an array of IPDRX records for every entity.
        - Type: String
        - Currently, supported values are ‘daily’, ‘hourly’ and ‘monthly’.

6.2 ftpUse

This synchronous call is used to extract usage information from the Flow Analyze and then post it to an FTP server. Ensure that the Flow Analyze has connectivity to the FTP server and the provided username/password have write access to the specified folder and file. If the file already exists, it will be overwritten.

This is a synchronous call and will return when the operation completed, with either a confirmation or an error message.

- Input
    - Entity Type
        - Type: String
        - The type of an entity. There are four types of entities in FA, which are ‘endpoint’, ‘application’, ‘interface’ and ‘location’. Therefore, users can only provide the call with these four types of string.
        - This parameter is mandatory.

    - Entity ID
        - Type: String
        - The identification of an entity. This is the equivalence of the Endpoint parameter in getIPDR call if entity type is ‘endpoint’.

    - Start Time
        - Type: Date
        - The UTC time from which usage information is requested. Actual returned time may be earlier as it starts at the boundary of the current interval. The start time is inclusive.

    - End Time

        - Type: Date
        - The UTC time up to which usage information is requested. Actual returned time may be later as it ends at the boundary of the current interval. The end time is exclusive.

    - Interval
        - Type: String
        - The valid values are 'daily', 'hourly' and 'monthly'.

    - Dimension
        - Type: String
        - The dimension of result data set. This parameter is used to separate the result data set in accordance with different entity types. If the user provisions 'application' as a dimension, the result data set will be separated by different applications running on the entity.
    - Hostname
        - Type: String
        - Either host or IP of the FTP server
    - Username
        - Type: String
        - FTP Username
    - Password
        - Type: String
        - FTP Password
    - Filename
        - Type:String
        - FTP Filename. If only a name is specified, the file will be placed in the home folder. Alternatively, you can specify one or more folders which will be created if it doesn't exist like dir1/dir2/filename. The file will be overwritten if it already exists.

- Output
    - This function returns an array of IPDRX records for every endpoint
        - Type: String
        - Currently, supported values are 'daily', 'hourly' and 'monthly'.

6.3 EndPointQuery

This synchronous call is used to search for FA endpoint objects based on endpoint name or IP address with substring case insensitive match. We provide 4 method:

1) search(String filter, Integer pageNumber, Integer pageSize) - filter is the search string used for substring case insensitive match with FA endpoint name or IP address. pageNumber and pageSize are used to support pagination. The search should return a list of endpoint objects which have the

following information for each endpoint object: IP address, endpoint name, mapping method, last updated date/time, customer email address, login username and password.

2) count(String filter)- return the total number of FA endpoints match with search filter

Notice1:

filter support partly match, it means filter can be either part of IP address or part of endpoint name.

3) getByIpaddress(String ipAddress) - return single endpoint object contains attributes listed in 1).

4) get(String pd_id) - return single endpoint object by internal prot_dom name - for example:

pd_id: "95945896_PD_441".

Notice2:

ipAddress& pd_id are full match, not support partly match. It means you must input whole information of them.

6.4 IPDRX

This record, which extends IPDR record, represents usage information for a given entity for a specified period.

Field	**Type**	**Extra**
SubscriberID	String	Full name of the entity.
SeqNum	double	A unique number assigned by interface for this request.
CreationTime	Date	UTC date when the IPDRX record is created.
DestAddress	String	Unused.
StartTime	Date	UTC date of the start boundary for the interval.
EndTime	Date	UTC date of the end boundary for the interval.
InputOctets	double	Bytes in to entity.
OutputOctets	double	Bytes out from entity.
avg_rate_up	double	Average rate out from entity.
avg_rate_down	double	Average rate in to entity.

maxOutputRate	double	Max rate out from entity.
maxInputRate	double	Max rate in to the entity.
toDim	String	The dimension of result data set.
mappingType	String	The mapping method used to obtain the endpoint name

6.5 WSDL

6.5.1 Usage.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://soap.v1.usage.fa.compass.calix.com"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://soap.v1.usage.fa.compass.calix.com"
xmlns:intf="http://soap.v1.usage.fa.compass.calix.com"
xmlns:tns1="http://data.soap.v1.usage.fa.compass.calix.com"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
 <wsdl:types>
  <schema elementFormDefault="qualified"
targetNamespace="http://soap.v1.usage.fa.compass.calix.com"
xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://data.soap.v1.usage.fa.compass.calix.com"/>
   <element name="ftpUse">
    <complexType>
     <sequence>
      <element name="entityType" type="xsd:string"/>
      <element name="entityId" type="xsd:string"/>
      <element name="startTime" type="xsd:date"/>
      <element name="endTime" type="xsd:date"/>
      <element name="interval" type="xsd:string"/>
      <element name="dimension" type="xsd:string"/>
      <element name="mappingDetail" type="xsd:boolean"/>
      <element name="hostname" type="xsd:string"/>
      <element name="username" type="xsd:string"/>
      <element name="password" type="xsd:string"/>
      <element name="filename" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="ftpUseResponse">
    <complexType/>
   </element>
   <element name="ftpIPDR">
    <complexType>
     <sequence>
      <element name="endpoint" type="xsd:string"/>
      <element name="startTime" type="xsd:dateTime"/>
      <element name="endTime" type="xsd:dateTime"/>
      <element name="interval" type="xsd:string"/>
      <element name="hostname" type="xsd:string"/>
      <element name="username" type="xsd:string"/>
      <element name="password" type="xsd:string"/>
      <element name="filename" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="ftpIPDRResponse">
    <complexType/>
   </element>
   <element name="getIPDR">
    <complexType>
     <sequence>
      <element name="endpoint" type="xsd:string"/>
      <element name="startTime" type="xsd:dateTime"/>
      <element name="endTime" type="xsd:dateTime"/>
      <element name="interval" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="getIPDRResponse">
    <complexType>
     <sequence>
      <element maxOccurs="unbounded" name="getIPDRReturn" type="tns1:IPDR"/>
     </sequence>
    </complexType>
   </element>
   <element name="getUse">
    <complexType>
     <sequence>
      <element name="entityType" type="xsd:string"/>
      <element name="entityId" type="xsd:string"/>
      <element name="startTime" type="xsd:date"/>
      <element name="endTime" type="xsd:date"/>
      <element name="interval" type="xsd:string"/>
      <element name="dimension" type="xsd:string"/>
      <element name="mappingDetail" type="xsd:boolean"/>
     </sequence>
    </complexType>
   </element>
   <element name="getUseResponse">
    <complexType>
     <sequence>
      <element maxOccurs="unbounded" name="getUseReturn" type="tns1:IPDRX"/>
     </sequence>
    </complexType>
   </element>
  </schema>
  <schema elementFormDefault="qualified"
targetNamespace="http://data.soap.v1.usage.fa.compass.calix.com"
xmlns="http://www.w3.org/2001/XMLSchema">
```

```
  <complexType name="IPDR">
   <sequence>
    <element name="avgInputRate" type="xsd:double"/>
    <element name="avgOutputRate" type="xsd:double"/>
    <element name="creationTime" nillable="true" type="xsd:dateTime"/>
    <element name="endTime" nillable="true" type="xsd:dateTime"/>
    <element name="inputOctets" type="xsd:double"/>
    <element name="outputOctets" type="xsd:double"/>
    <element name="seqNum" type="xsd:long"/>
    <element name="startTime" nillable="true" type="xsd:dateTime"/>
    <element name="subscriberID" nillable="true" type="xsd:string"/>
   </sequence>
  </complexType>
  <complexType name="IPDRX">
   <complexContent>
    <extension base="tns1:IPDR">
     <sequence>
      <element name="mappingType" nillable="true" type="xsd:string"/>
      <element name="maxInputRate" type="xsd:double"/>
      <element name="maxOutputRate" type="xsd:double"/>
      <element name="toDim" nillable="true" type="xsd:string"/>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
 </schema>
</wsdl:types>

  <wsdl:message name="ftpIPDRResponse">

     <wsdl:part element="impl:ftpIPDRResponse" name="parameters">

     </wsdl:part>

  </wsdl:message>

  <wsdl:message name="ftpUseRequest">

     <wsdl:part element="impl:ftpUse" name="parameters">

     </wsdl:part>

  </wsdl:message>

  <wsdl:message name="getUseRequest">

     <wsdl:part element="impl:getUse" name="parameters">

     </wsdl:part>

  </wsdl:message>

  <wsdl:message name="getIPDRResponse">
```

```
   <wsdl:part element="impl:getIPDRResponse" name="parameters">

   </wsdl:part>

</wsdl:message>

<wsdl:message name="ftpIPDRRequest">

   <wsdl:part element="impl:ftpIPDR" name="parameters">

   </wsdl:part>

</wsdl:message>

<wsdl:message name="getUseResponse">

   <wsdl:part element="impl:getUseResponse" name="parameters">

   </wsdl:part>

</wsdl:message>

<wsdl:message name="getIPDRRequest">

   <wsdl:part element="impl:getIPDR" name="parameters">

   </wsdl:part>

</wsdl:message>

<wsdl:message name="ftpUseResponse">

   <wsdl:part element="impl:ftpUseResponse" name="parameters">

   </wsdl:part>

</wsdl:message>

<wsdl:portType name="Usage">

   <wsdl:operation name="ftpUse">

      <wsdl:input message="impl:ftpUseRequest" name="ftpUseRequest">

    </wsdl:input>

      <wsdl:output message="impl:ftpUseResponse" name="ftpUseResponse">

    </wsdl:output>

   </wsdl:operation>

   <wsdl:operation name="ftpIPDR">
```

```
         <wsdl:input message="impl:ftpIPDRRequest" name="ftpIPDRRequest">

       </wsdl:input>

         <wsdl:output message="impl:ftpIPDRResponse" name="ftpIPDRResponse">

       </wsdl:output>

      </wsdl:operation>

      <wsdl:operation name="getIPDR">

         <wsdl:input message="impl:getIPDRRequest" name="getIPDRRequest">

       </wsdl:input>

         <wsdl:output message="impl:getIPDRResponse" name="getIPDRResponse">

       </wsdl:output>

      </wsdl:operation>

      <wsdl:operation name="getUse">

         <wsdl:input message="impl:getUseRequest" name="getUseRequest">

       </wsdl:input>

         <wsdl:output message="impl:getUseResponse" name="getUseResponse">

       </wsdl:output>

      </wsdl:operation>

   </wsdl:portType>

   <wsdl:binding name="UsageSoapBinding" type="impl:Usage">

      <wsdlsoap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>

      <wsdl:operation name="ftpUse">

         <wsdlsoap:operation soapAction=""/>

         <wsdl:input name="ftpUseRequest">

            <wsdlsoap:body use="literal"/>

         </wsdl:input>

         <wsdl:output name="ftpUseResponse">

            <wsdlsoap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="ftpIPDR">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="ftpIPDRRequest">
            <wsdlsoap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="ftpIPDRResponse">
            <wsdlsoap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getIPDR">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="getIPDRRequest">
            <wsdlsoap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="getIPDRResponse">
            <wsdlsoap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getUse">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="getUseRequest">
            <wsdlsoap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="getUseResponse">
            <wsdlsoap:body use="literal"/>
         </wsdl:output>

      </wsdl:operation>

   </wsdl:binding>

   <wsdl:service name="UsageService">

      <wsdl:port binding="impl:UsageSoapBinding" name="Usage">

         <wsdlsoap:address
location="http://localhost:8080/soap/services/Usage"/>

      </wsdl:port>

   </wsdl:service>

</wsdl:definitions>
```

6.5.2 EndpointQuery.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://epquery.fa.compass.calix.com"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://epquery.fa.compass.calix.com"
xmlns:intf="http://epquery.fa.compass.calix.com"
xmlns:tns1="http://data.epquery.fa.compass.calix.com"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
 <wsdl:types>
  <schema elementFormDefault="qualified"
targetNamespace="http://epquery.fa.compass.calix.com"
xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://data.epquery.fa.compass.calix.com"/>
   <element name="get">
    <complexType>
     <sequence>
      <element name="pd_id" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="getResponse">
    <complexType>
     <sequence>
      <element name="getReturn" type="tns1:EndpointData"/>
     </sequence>
    </complexType>
   </element>
   <element name="count">
    <complexType>
```

```
     <sequence>
      <element name="filter" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="countResponse">
    <complexType>
     <sequence>
      <element name="countReturn" type="xsd:int"/>
     </sequence>
    </complexType>
   </element>
   <element name="search">
    <complexType>
     <sequence>
      <element name="filter" type="xsd:string"/>
      <element name="pageNumber" type="xsd:int"/>
      <element name="pageSize" type="xsd:int"/>
     </sequence>
    </complexType>
   </element>
   <element name="searchResponse">
    <complexType>
     <sequence>
      <element maxOccurs="unbounded" name="searchReturn"
type="tns1:EndpointData"/>
     </sequence>
    </complexType>
   </element>
   <element name="getByIpAddress">
    <complexType>
     <sequence>
      <element name="ipAddress" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="getByIpAddressResponse">
    <complexType>
     <sequence>
      <element name="getByIpAddressReturn" type="tns1:EndpointData"/>
     </sequence>
    </complexType>
   </element>
  </schema>
  <schema elementFormDefault="qualified"
targetNamespace="http://data.epquery.fa.compass.calix.com"
xmlns="http://www.w3.org/2001/XMLSchema">
   <complexType name="EndpointData">
    <sequence>
     <element name="disName" nillable="true" type="xsd:string"/>
     <element name="id" nillable="true" type="xsd:string"/>
     <element name="ipaddress" nillable="true" type="xsd:string"/>
     <element name="lastUpdateTime" type="xsd:long"/>
     <element name="mappingMethod" nillable="true" type="xsd:string"/>
     <element name="uid" nillable="true" type="xsd:string"/>
     <element name="userEmail" nillable="true" type="xsd:string"/>
     <element name="userName" nillable="true" type="xsd:string"/>
     <element name="userPassword" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
  </schema>
 </wsdl:types>

   <wsdl:message name="getByIpAddressRequest">

      <wsdl:part element="impl:getByIpAddress" name="parameters">

      </wsdl:part>

   </wsdl:message>

   <wsdl:message name="searchRequest">

      <wsdl:part element="impl:search" name="parameters">

      </wsdl:part>

   </wsdl:message>

   <wsdl:message name="getByIpAddressResponse">

      <wsdl:part element="impl:getByIpAddressResponse" name="parameters">

      </wsdl:part>

   </wsdl:message>

   <wsdl:message name="searchResponse">

      <wsdl:part element="impl:searchResponse" name="parameters">

      </wsdl:part>

   </wsdl:message>

   <wsdl:message name="countResponse">

      <wsdl:part element="impl:countResponse" name="parameters">

      </wsdl:part>

   </wsdl:message>

   <wsdl:message name="countRequest">

      <wsdl:part element="impl:count" name="parameters">

      </wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="getRequest">
   <wsdl:part element="impl:get" name="parameters">
   </wsdl:part>
</wsdl:message>
<wsdl:message name="getResponse">
   <wsdl:part element="impl:getResponse" name="parameters">
   </wsdl:part>
</wsdl:message>
<wsdl:portType name="EndpointQuery">
   <wsdl:operation name="get">
      <wsdl:input message="impl:getRequest" name="getRequest">
    </wsdl:input>
      <wsdl:output message="impl:getResponse" name="getResponse">
    </wsdl:output>
   </wsdl:operation>
   <wsdl:operation name="count">
      <wsdl:input message="impl:countRequest" name="countRequest">
    </wsdl:input>
      <wsdl:output message="impl:countResponse" name="countResponse">
    </wsdl:output>
   </wsdl:operation>
   <wsdl:operation name="search">
      <wsdl:input message="impl:searchRequest" name="searchRequest">
    </wsdl:input>
      <wsdl:output message="impl:searchResponse" name="searchResponse">
    </wsdl:output>
```

```
      </wsdl:operation>

      <wsdl:operation name="getByIpAddress">

         <wsdl:input message="impl:getByIpAddressRequest"
name="getByIpAddressRequest">

       </wsdl:input>

         <wsdl:output message="impl:getByIpAddressResponse"
name="getByIpAddressResponse">

       </wsdl:output>

      </wsdl:operation>

   </wsdl:portType>

   <wsdl:binding name="EndpointQuerySoapBinding" type="impl:EndpointQuery">

      <wsdlsoap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>

      <wsdl:operation name="get">

         <wsdlsoap:operation soapAction=""/>

         <wsdl:input name="getRequest">

            <wsdlsoap:body use="literal"/>

         </wsdl:input>

         <wsdl:output name="getResponse">

            <wsdlsoap:body use="literal"/>

         </wsdl:output>

      </wsdl:operation>

      <wsdl:operation name="count">

         <wsdlsoap:operation soapAction=""/>

         <wsdl:input name="countRequest">

            <wsdlsoap:body use="literal"/>

         </wsdl:input>

         <wsdl:output name="countResponse">
```

```
            <wsdlsoap:body use="literal"/>

         </wsdl:output>

      </wsdl:operation>

      <wsdl:operation name="search">

         <wsdlsoap:operation soapAction=""/>

         <wsdl:input name="searchRequest">

            <wsdlsoap:body use="literal"/>

         </wsdl:input>

         <wsdl:output name="searchResponse">

            <wsdlsoap:body use="literal"/>

         </wsdl:output>

      </wsdl:operation>

      <wsdl:operation name="getByIpAddress">

         <wsdlsoap:operation soapAction=""/>

         <wsdl:input name="getByIpAddressRequest">

            <wsdlsoap:body use="literal"/>

         </wsdl:input>

         <wsdl:output name="getByIpAddressResponse">

            <wsdlsoap:body use="literal"/>

         </wsdl:output>

      </wsdl:operation>

   </wsdl:binding>

   <wsdl:service name="EndpointQueryService">

      <wsdl:port binding="impl:EndpointQuerySoapBinding"
name="EndpointQuery">

         <wsdlsoap:address
location="http://localhost:8080/soap/services/EndpointQuery"/>

      </wsdl:port>
   </wsdl:service>

</wsdl:definitions>
```