CHAPTER 1

Getting Started

This chapter provides an overview of XML-RPC and includes the following sections:

Prerequisites

This document is intended for Cisco partners and customers who wish to integrate their products with Cisco Network Building Mediator. This document assumes the reader is familiar with the Cisco Network Building Mediator Release 3.1.1 and Cisco Network Building Mediator Manager Release 1.1.1 product and has a working knowledge of XML, web services, and Python.

Overview

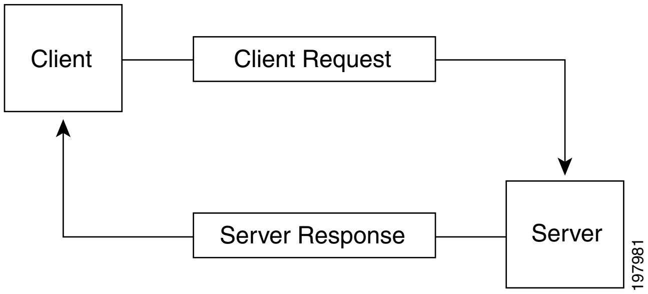
The Cisco Network Building Mediator (Mediator) provides an XML-RPC Application Programming Interface (API) that enables third-party software applications to communicate with the Mediator and to consume the data and services offered by the Mediator.

The API allows the application to:

- Interact with the Mediator framework
  - Start, stop, and query nodes in the framework
  - Read/write node values in the framework
- Communicate with remote Mediators using RNA over XML-RPC.
- Integrate with several special services on the Mediator such as:
  - Subscription Manager (allows you to subscribe to a set of points that may be polled in a batch)
  - Logger (The Logger service lets you query and log values from the framework)
- User authentication is built into the API to control unauthorized access to nodes and services in the Mediator.

XML-RPC is a simple protocol used for carrying out remote procedure calls (RPC) over TCP/IP. It uses two Internet standards, Hypertext Transfer Protocol (HTTP) and eXtensible Markup Language (XML) to create a standard method of calling remote web services and receiving a response (see Figure 1-1).

Figure 1-1 XML-RPC Diagram



For more information about the XML-RPC standard, go to http://www.xmlrpc.com. For more information about the XML-RPC support on Python, go to http://docs.python.org/library/xmlrpclib.html#module-xmlrpclib**.**

After reading this guide, you should understand the following:

- Standard APIs provided by Mediator framework nodes useful to RPC clients.
- XML-RPC and how the Mediator framework supports it.
- RNA and how to make use of it via XML-RPC.
- Specialized APIs provided by services useful to RPC clients.

XML-RPC Handlers

The XML-RPC support on the Mediator is provided by an HTTP request handler named XMLRPCv2. This request handler is installed and enabled under both the HTTP and HTTPS servers of every Mediator and Mediator Manager by default.

The Mediator supports the following two versions of XML-RPC service:

- XML-RPCv1
- XML-RPCv2

The key features of the two handlers are summarized as follows:

1.XML-RPCv1 supports explicit use of username and password, via invoking a method.

2.XML-RPCv1 uses a session identifier (SID) to track authorization. This SID is returned by the create-session method, and is required as the first parameter to all other invocations.

3.XML-RPCv2 requires BASIC authentication. Clients must provide valid authorization headers with their requests in order for those requests to be authorized.

4.XML-RPCv2 does not require SID.

Both handlers have subservices that support specific applications. The primary subservice of interest is the Remote Node Abstraction (RNA) subservice. RNA enables clients to easily interact with Mediator framework nodes. Via RNA, clients may query and manipulate nodes, retrieve and override values, and so on.

XML-RPC Based Interaction

This section describes the XML-RPC-based interaction with the Mediator framework and framework nodes using XML-RPCv2. The examples provided in this guide show interaction with the framework using the Python standard XML-RPC library from the Python prompt. Output of the HTTP(S) exchange is provided where needed for clarification. XML-RPC service is provided via HTTP and HTTPS request handlers. These handlers are invoked based on the request path, or URL, of incoming requests. Standard framework configurations set up the XML-RPCv2 handlers for handling requests to any path beginning with /XMLRPCv2.

You are encouraged to try the examples shown here, or to experiment with the Mediator XML-RPC support. The examples provided require that Python be installed. You can verify Python support on your system by typing Python at a command prompt. If Python exists on the system, then the command will execute and launch the Python interactive prompt. If Python is not installed, you may install Python by downloading it from www.python.org and following the installation instructions appropriate for their OS.

Remote Node Abstraction (RNA) is a subservice of the XML-RPC handlers. The RNA subservice is selected by appending /RNA to the request path. A normal XML-RPCv2 RNA request for URLs begin with /XMLRPCv2/RNA. When using RNA, clients specify the framework node to which a request should be applied by appending that node URL to the base RNA over XML-RPCv2 URL. For example, to interact with the node whose node-URL is /interfaces/relay1, the client sends a request for the URL /XMLRPCv2/RNA/interfaces/relay1.

By using the request URL to specify the node or service that an XML-RPC request should be applied to, you essentially set the namespace within which their XML-RPC invocation will be resolved. Any method provided by the targeted node may now be invoked via standard XML-RPC method call.

XML-RPCv2 requires BASIC authentication for authorization. Most XML-RPC libraries provide support for BASIC authentication directly. In the event that a library does not support BASIC authentication, you need to add the authorization header to their XML-RPC requests explicitly.