



# CHAPTER 1

## Introduction

---

This chapter provides a detailed overview of Building Automation System (BAS), the Cisco Network Building Mediator system, and Mediator Operating Environment (MOE) system software.

This chapter includes the following sections:

- [Building Automation System Overview, page 1-1](#)
- [Mediator System Overview, page 1-2](#)
- [Mediator Framework Overview, page 1-4](#)
- [Mediator Operating Environment \(MOE\) System Software, page 1-5](#)



**Note**

---

Before you install, operate, or service the system, read the *Regulatory Compliance and Safety Information for the Cisco Network Building Mediator* document for important safety information.

---

## Building Automation System Overview

Building Automation System (BAS) describes the functionality of the control system, which offers comprehensive and co-coordinated control of one or more major system functions required in a facility, such as the following:

- Heating, Ventilating, and Air Conditioning (HVAC) system
- Lighting system
- Security system
- Fire and Life safety system

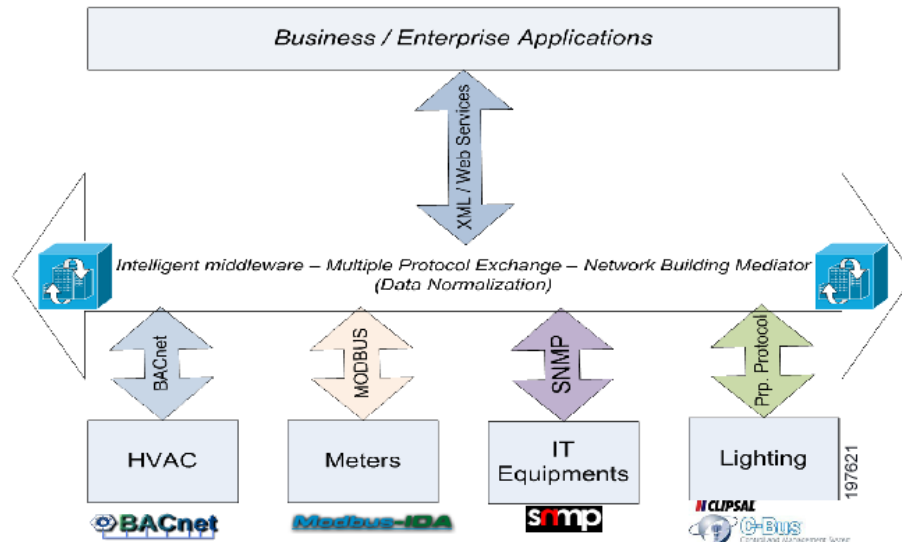
These building systems are still being installed and operated in silos for either security reasons or due to the challenges in system integration, but this reliance limits the types of applications that can emerge from the integration. In particular, it limits your ability to use BAS for your own purposes, limiting the ability to integrate BAS functions with real enterprise systems. The move to more integrated building services also requires tighter integration of the mostly heterogeneous technologies. Here evolves the need for an intelligent middleware technology that can enable businesses to integrate building infrastructure and information technology (IT) applications over a common Internet Protocol (IP) network. This intelligent middleware would aggregate, normalize, make available all data from the siloed building systems, and allow data to pass bi-directionally from the building systems to enterprise business applications through web services.

**Send documentation comments to [cbsbu-docfeedback@cisco.com](mailto:cbsbu-docfeedback@cisco.com)**

The core function of the BAS is to keep the building climate within a specified range, provide lighting based on an occupancy schedule, monitor system performance and device failures, and provide e-mail or text notifications to building engineering staff. The BAS functionality reduces building energy and maintenance costs when compared to a non-controlled building.

Figure 1-1 displays a typical Cisco Network Building Mediator(Mediator) integration architecture.

**Figure 1-1 Cisco Network Building Mediator Integration Architecture**



## Mediator System Overview

The Mediator provides a network-based framework, which allows the convergence of multiple disparate building systems onto an IP network, resulting in improved efficiency and greater energy savings.

A Mediator system includes hardware and software components. The Mediator is a hardware component; configTOOL, perfectHOST, and Mediator web client are the software components of the system.

- configTOOL—The software used to configure the Multi-Protocol Exchange (MPX) of the Mediator.
- perfectHOST—The graphical programming software used to program the Cisco Building Intelligence controllers.
- Mediator web client—A suite of advanced system software tools used to configure, program, and monitor the Mediator.



**Warning**

**Installation of the equipment must comply with local and national electrical codes.**

Figure 1-2 displays a typical Mediator system architecture.

Send documentation comments to [cbsbu-docfeedback@cisco.com](mailto:cbsbu-docfeedback@cisco.com)

Figure 1-2 Mediator System Architecture

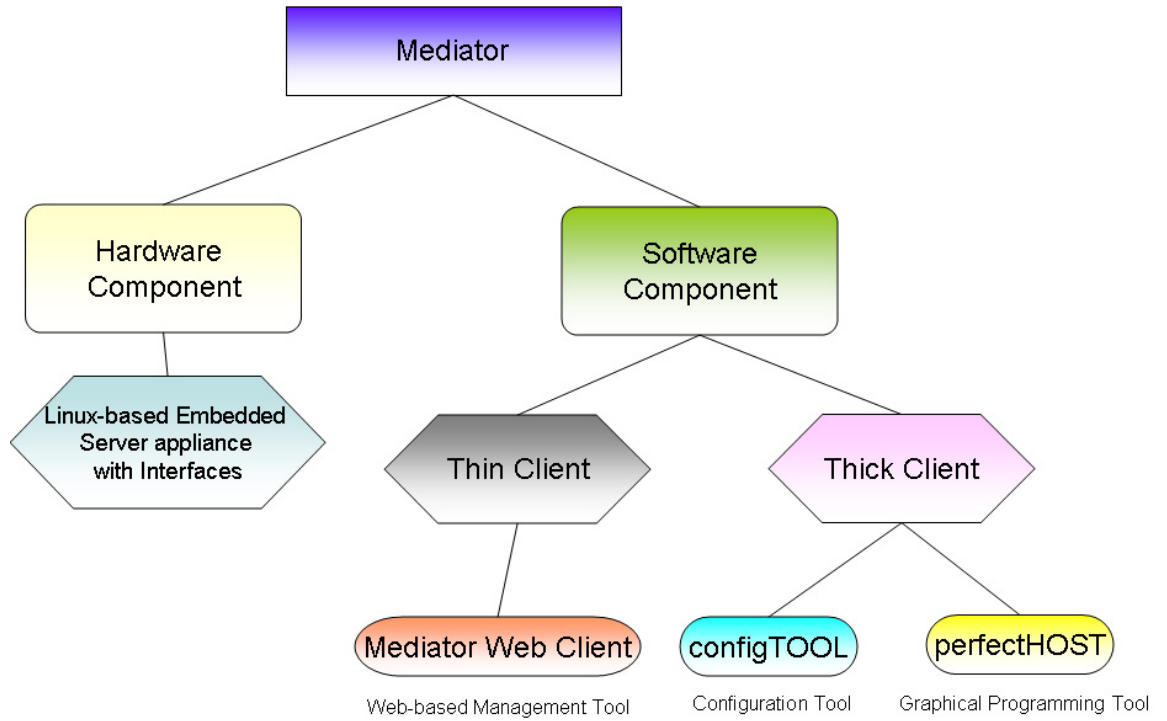
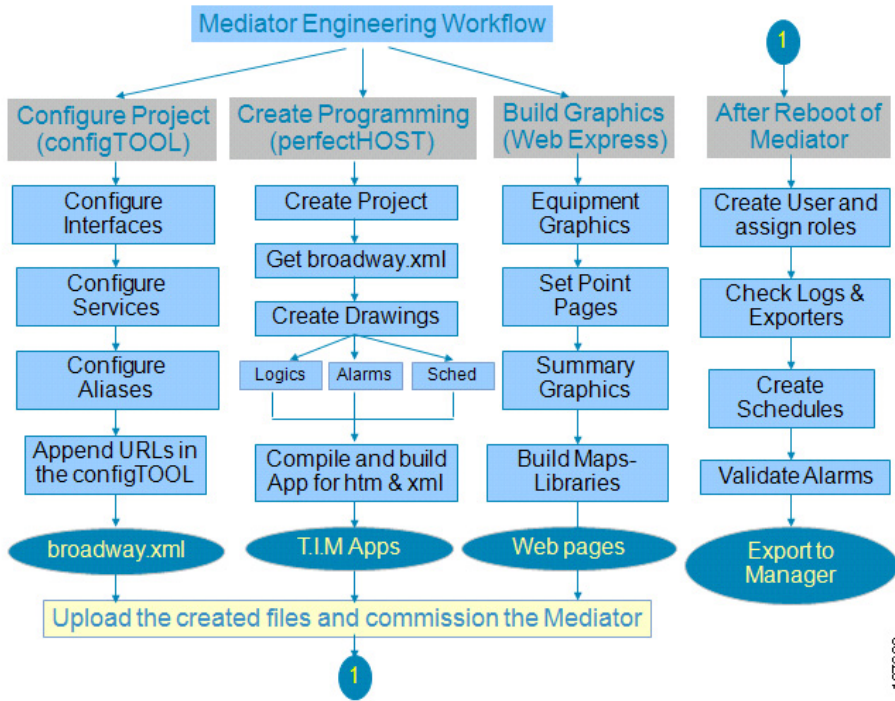


Figure 1-3 displays a typical Mediator engineering workflow.

Figure 1-3 Mediator Engineering Workflow



197638

[Send documentation comments to cbsbu-docfeedback@cisco.com](mailto:cbsbu-docfeedback@cisco.com)

# Mediator Framework Overview

Mediator Framework is the software application that provides the core function of the Mediator. The Mediator Framework gives the Mediator its Multi-Protocol Exchange capabilities, including the following:

- Assembling a running system from configuration data.
- Getting and setting values.
- Processing raw data.
- Presenting data from disparate sources in a uniform format.
- Making data easily available to you.
- Providing services such as data logging and HTTP services.
- Providing an environment for custom system configuration.
- Providing an environment for developing custom solutions.

A key feature of the Mediator Framework architecture is Dynamic System Configuration (DSC). DSC allows the Mediator Framework to assemble a system that includes only the program code required by a particular site configuration.

The Mediator Framework provides the structure upon which represented site systems are built. It provides the following functionality, allowing the Mediator to configure any site:

- Assembly of the system from configuration data.
- The operating environment in which that system runs.
- The existing framework can be modified or extended without shutting the system down.

The Mediator Framework represents a site by assembling a tree structure of objects called nodes. Nodes are software abstractions that represent the relationships between objects in the system. A node is a named object that exchanges information with other nodes.

For example, an RS-232 port, a Modbus line handler, a data point, and so on are named objects which support the exchange of information with other nodes.

The Mediator Framework structures nodes hierarchically. In the Mediator Framework terminology, a child node is a node that has a subordinate relationship to another node (for example, a Modbus handler configured under a COM port). The node to which a child is subordinate is called a parent node. The child nodes that the Mediator Framework automatically inserts under the parent nodes are called inherent child nodes.

The Mediator Framework creates a global name space on the Mediator in the form of a URL structure that resembles a file path made up of objects with parent-child relationships.

For example, the file:

```
/interfaces/com1/modbus/meter_1/max.kW
```

uniquely identifies a data point *max.kW* as a node whose parent node is a Veris meter personality, whose parent node is a Modbus line handler, whose parent node is com1, whose parent node is the interfaces anchor, whose parent node is the host device.

***Send documentation comments to [cbsbu-docfeedback@cisco.com](mailto:cbsbu-docfeedback@cisco.com)***

The top level (root) of the node tree is the host, and is represented by a forward slash (/). In the standard system, the host is a Mediator device. Because every node has a name and a parent going up to the root node in the name space, the Mediator Framework can look up any node on the Mediator host by its URL name. The nodes at the level immediately under the root are called anchors. The configTOOL has three anchor nodes representing the three main node categories of the Mediator Framework:

1. Interfaces—Interface nodes include the I/O ports and sensors of the Mediator, and the devices connected to them.
2. Services—Service nodes support data exchange (for example, the Logger, network services such as the HTTP server, and so on.)
3. Aliases—Alias nodes are used to assign alternative names to other nodes. Aliases allow you to assign a meaningful name to any node in the system. In addition, you can use aliases to group into a single folder related nodes that reside in different parts of the node tree.

The Mediator software includes two files that determine the operating parameters:

- The configuration file
- The configuration database

The configuration file is an XML file that describes the system configuration. The configTOOL generates this file after you finish and saves the entries specifying the configuration. At system startup, the Mediator Framework parses the configuration file and uses the information it contains to configure all of the nodes in the system.

**Note**

The standard Mediator software includes a generic configuration file that allows the Mediator Framework to configure a minimal running system if configTOOL is not used to generate a configuration file.

The configuration database is an XML file that contains the NodeDefs (the node definitions) that specify the capabilities and requirements for each node. When you configure a node, configTOOL uses the information in the configuration database file to list the available options based on matching parent node capabilities to the requirements of child node.

The configTOOL allows you to edit the configuration file.

## Mediator Operating Environment (MOE) System Software

The Mediator Operating Environment (MOE) system software is based on the current Linux kernel. Linux is a UNIX-like, open-source, fully featured operating system that provides true multitasking, multithreading, virtual memory, shared libraries, demand loading, shared copy-on-write executables, sophisticated memory management, loadable device driver modules, and TCP/IP networking. It allows the Mediator to run multiple services and protocols simultaneously and still respond to events generated in the network environment. Linux supports a broad range of open and secure networking capabilities.

In the event of a system failure, its full-journaling file system restores the data on the disk to its pre-failure configuration. It also recovers unsaved data and stores it in the location where it would have gone if the system had not failed.

The Mediator software consists of two main components, the MOE (operating system) and the Mediator Framework (application).

***Send documentation comments to [cbsbu-docfeedback@cisco.com](mailto:cbsbu-docfeedback@cisco.com)***