

Network Applications of Savoy Servers

Savoy's Server module is a Win98/NT/CE application designed for monitoring and control of complex devices. Inherent within the module is the ability to combine them with other modules in a potentially large network. This document describes many of the networking features of the Savoy module and how they can be applied to solving real problems.

The general capabilities of the Savoy module including Rule Engine, Java Scripting, Device/Domain design, Capability access control, can be found in a separate document entitled 'Architecture of the Savoy Automation System'. This document assumes familiarity with these concepts and focussed primarily on networking issues.

Devices and their Proxies

The principle entity with the Savoy system is the 'device', so named because much of what the Server does relates to actual real-world devices. However, the concept of a 'device' is considerably broader and includes many of the data types found in other programming environments. These are termed 'logical devices' in Savoy and include Boolean, Strings, and a variety of others. Devices, then, represent the fundamental 'name space' of the system and the capabilities of the devices represent the total capability of the server.

When two or more servers interconnect, they exchange device information and create 'proxy' devices that represent devices located on a connected server. This has two implications:

1. All devices across the network are in the same address space
2. When connected, each server has access to all devices, regardless of where they are.

Assertions to Devices Propagate across the Network

Assertions are changes of device state that originate from devices, rule engine(s), Java scripts, client applications, and so on.

Since assertions are automatically routed across the network, rules can be written to run in any server and control devices on any other server.

When authoring rules, it is recommended that the 'cause' side of the rules be on the server that hosts the referenced device, whereas the 'effect' side of the rule can apply to any device, local or not. This restriction does not apply to logical devices.

Scalability and Name Scoping

The concept of creating proxies for all remote devices has the obvious limitation of growing to very large numbers -- roughly as the square of the number of servers.

To prevent this limitation and permit the number of interconnected server to grow arbitrarily large, Savoy has implemented a concept of 'name scoping'. Name scoping is similar to that found in programming languages and is designed to contain the name of a device within a certain 'scope'. The Savoy Server implements a very general mechanism that functions as follows:

1. Each device is assigned a Scope Name -- a simple string, like 'Campus', or 'Building3'.
2. Each Server has a list of Scope Names.

3. Proxies are created on Servers ONLY IF that server possesses the device's Scope Name.

Imagine a network of Servers arranged on a map, and then think of a Scope as being singly connected area on the map. Singly connected means that there should only be a single area of any given name, not two or more areas.

There can be many Scopes and they can overlap arbitrarily. You can arrange them hierarchically (one Scope being totally contained within another) or not (intersecting Scopes).

To permit large networks, devices having broad scope should be few in number. Devices having narrow scope can be large in number. In other words, within any given Server, assign scope names so that most devices have local scope, a small percentage has broader scope, and only a few have global scope, for example.

Connecting Servers

A network of Servers is configured by two simple techniques:

1. Each Savoy Server can connect to a single parent.
3. Each Savoy Server can be the parent of many children.

Servers can connect to parents by configuration using the Setup dialog, or it can be done dynamically via rules. The child always initiates connections, and so each Server need only know the IP Address of its parent.

It should be emphasized that the terms 'parent' and 'child' refer to the topology of interconnect and conveys no meaning as to how the Servers interact. In fact, once connected, all Servers interact as peers.

Connections are persistent and reliable. For example, individual Servers can be shut down momentarily and the network is automatically re-established on power up.

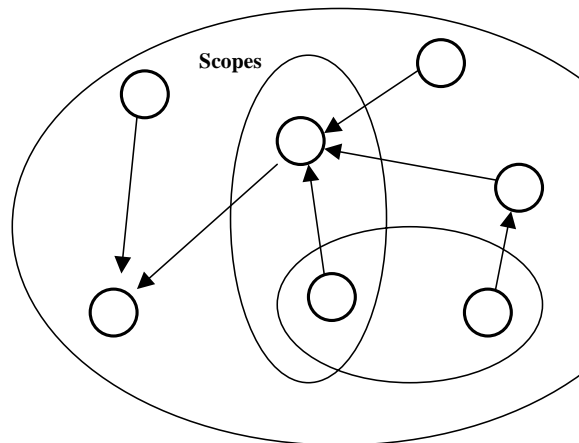


Figure 1 Networked Servers showing Topology and Name Scopes

Client Applications

Savoy provides a variety of client applications together with software components that permit easy development of customer applications. In general, client applications can connect to any Server and see all devices that are in-scope. Clients can also originate assertions to control devices and receive assertions to change the clients rendering of device state.