Policy Builder Overview

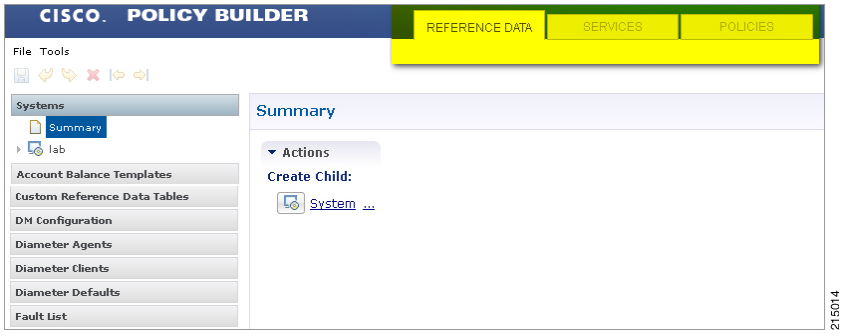
Overview

Cisco Policy Suite (CPS) provides a framework for building rules that can be used to enforce business logic against policy enforcement points such as network routers and packet data gateways. For example, a prepaid customer (one who pays as they go) might be denied service or prompted to top-up when their quota has expired, whereas a postpaid customer (one who has an ongoing billing relationship with the service provider) might only have their service downgraded or be automatically billed for additional data when their particular quota has expired.

CPS allows service providers to create policies that are customized to their particular business requirements through the use of the CPS Policy Builder, a web-based tool with a graphical user interface (GUI) that allows for rapid development of innovative new services.

The Policy Builder GUI supports both configuration of the overall CPS cluster of virtual machines (VMs) as well as the configuration of services and advanced policy rules. The following sections introduces the main aspects of the PB GUI as laid out in three tabs on the upper right of the interface: Reference Data, Services and Policies.

Figure 1: Cisco Policy Guilder GUI



Reference Data

The Reference Data tab of the PB GUI provides access for configuring various aspects of the system in order to make the system ready for operation. Reference Data are used to not only configure the system, but are also used to provide settings and parameters that are referenced by policy rules across various services; for example, Account Balances and Notifications are configured as Reference Data but are then referenced and reused by multiple services as needed. Details of the various Reference Data configuration options are described in more detail in other chapters of this guide.

The Reference Data tab contains static system, network, and template definition. It is not directly related to policy, services, or use cases, but does define the reference points for the following types of information:

- Systems, cluster, and instance data
- Jdbc query string definitions
- Balance and quota definitions
- Diameter agents, clients, and defaults information
- Query strings
- Custom reference data tables (custom look up tables such as apn names)
- Notification addresses and text templates
- Policy reporting criteria
- Subscriber data repositories
- Tariff switch times
- Fault list - For more information, refer to *CPS Operations Guide* for this release.

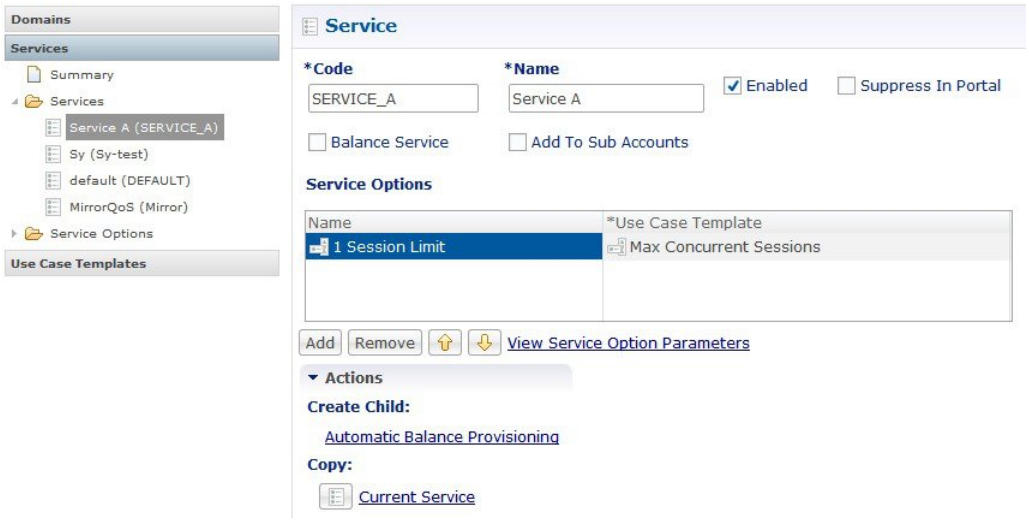
Services

The Services tab allows for creation of reusable policy rules that control how subscribers are granted network services, quota and notifications. Services are broken down into three core areas: Domains, Services and Use Case Templates. The following section provides an overview of the Services tab, however detailed instructions on how to build a service are covered in later chapters of this guide.

The creation of a new service begins with creating a Use Case Template (UCT) for the service. UCTs consist of Service Configurations specific to the service that will be created. For example, a Service Configuration might provide for the setup of a Gx Rule or Basic QoS. The UCT is also used to configure Use Case Initiators (UCI) which are instructions on when a specific Service Configuration should be in effect. An example of the UCI might be “only send this Gx Rule when the account balance is depleted”. Multiple UCIs can be configured for each Service Configuration allowing for complex logic as to when the configuration should or should not be in effect.

Once a UCT and associated UCIs are defined, it becomes the basis for Service Options, which are specific instances of the UCT that are populated with data specific to the service. Multiple Service Options can be created from a single UCT; for example, a UCT that provides for passing QoS parameters can be reused with different QoS values for different customers. Multiple Service Options can be layered to create the end Service.

Figure 2: Services tab



The Domains panel within the Services tab handles the initial interaction of the client device with the policy engine, and covers tasks including client authentication, default provisioning of unknown clients and qualifying a client for particular system defaults and services.

For more information on the Services tab, refer to the Services chapter.

Policies

While the Services tab, through Use Case Templates and Service Options, makes it easy to create reusable and extensible services, the Policies tab allows direct access to the underlying policy engine. The Policies tab holds the CPS core system Blueprint, which is composed of various Extension Points that break the policy engine flow into sections that occur within the execution of the policy. For example, the point in the policy flow where a Gx connection is received, parsed, and processed before the point in the policy flow where the related subscriber data is evaluated.

Within the various Extension Points are Policies that define Conditions (events and data from the policy flow and external systems) that can then trigger Actions (manipulation of data and communication back to external systems).

Note that the configuration of services for most deployments will be handled through use of the Reference Data and Services tabs; advanced policies as defined on the Policies tab and discussed above are only required for complex deployments. It is recommended that only experienced users access the Policies tab as errors in custom policies can have negative impact on the operation of the system. Detailed discussion of custom policies is outside of the scope of this document.

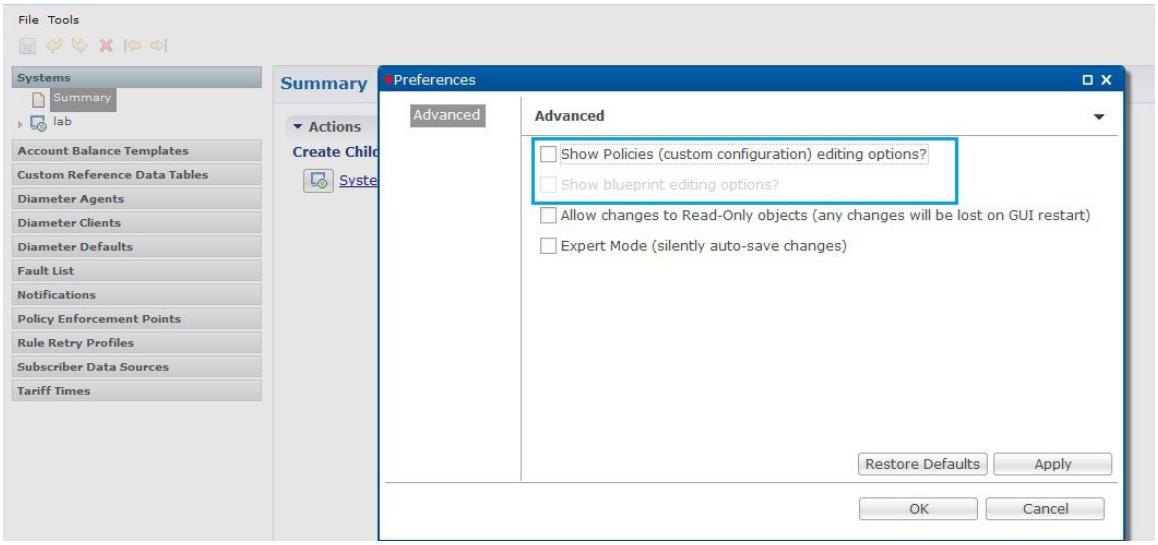
By default, the Policies and Blueprint tabs are disabled.

Important The Policy Builder offers the Blueprint section under **Policies** tab to enable Cisco recommended changes to the Policy Engine. Changes made without Cisco guidance are not supported and can result in poor performance, platform instability, or reduced capacity.

Enabling POLICIES tab

In Policy Builder, **Tools** > **Preferences** to open **Preferences** pop-up window.

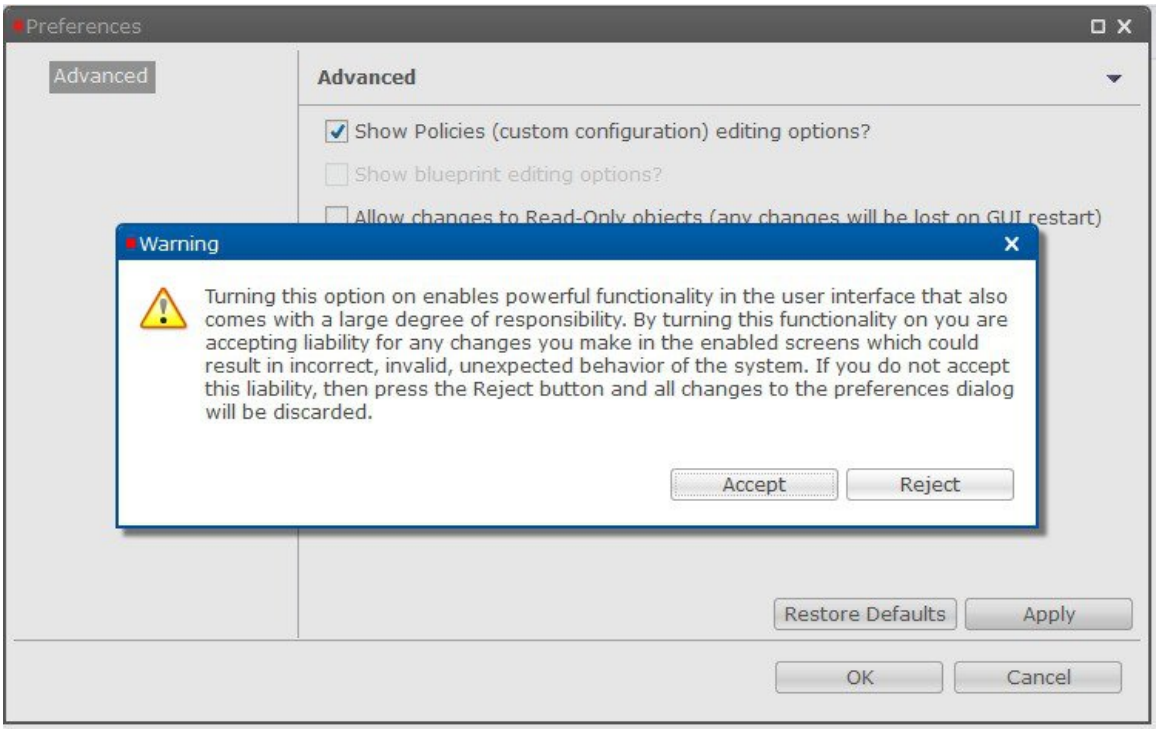
Figure 3: Preferences - Policies



Select Show Policies (custom configuration) editing options? and click **Apply**.

Warning pop-up dialog box opens up.

Figure 4: Warning



Click **Accept** so that **POLICIES** tab is visible in Policy Builder.

Enabling BLUEPRINTS tab

For **BLUEPRINTS** tab to be visible in Policy Builder, Show Policies (custom configuration) editing options? must be checked.

Select Show blueprint editing options? and click **Apply**.

Figure 5: Preferences - Blueprints 1

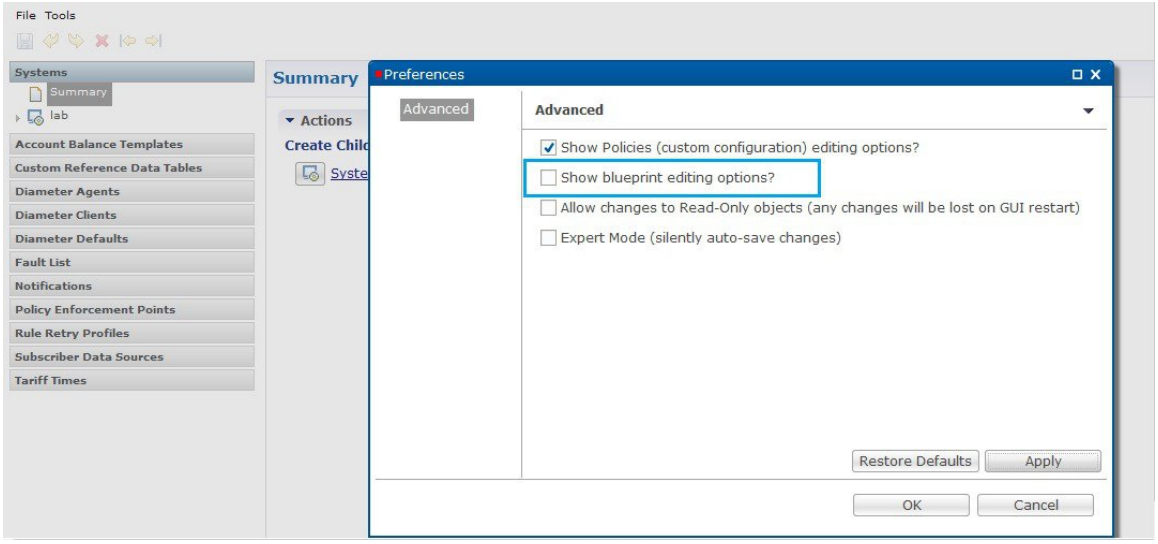
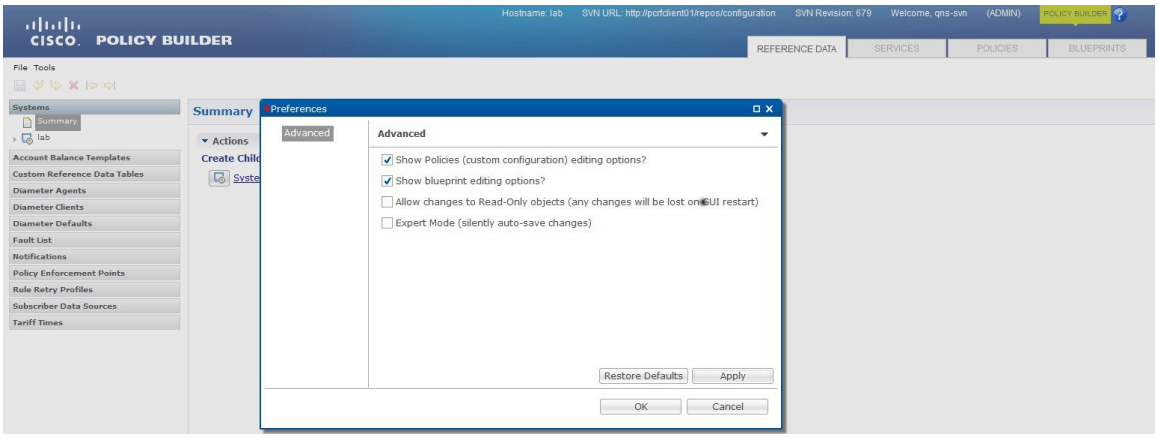


Figure 6: Preferences - Blueprints 2



Warning pop-up dialog box opens up.

Click **Accept** so that **BLUEPRINTS** tab is visible in Policy Builder.

Note In case the **POLICIES** checkbox is unchecked while **BLUEPRINTS** checkbox is checked, the **BLUEPRINTS** checkbox is unchecked forcefully and the **BLUEPRINTS** tab is not visible.

Summary of Policy Tab Capabilities

- Conditional rules within specified Extension Points (Condition/Action)
- Trigger specific actions from an extensive catalog of Use Case Initiators
- Evaluate and manipulate session data as part of making policy decisions and returning services data to downstream systems

Advantages

- Allows for handling complex policy situations without writing custom code
- Support for custom or unusual business rules

Considerations

- Building custom policies requires a deep understanding of the call flow and underlying CPS platform
- Due to the flexibility of the Policy Builder, it is possible to create conflicting policies that can have a negative impact on system performance

Accessing the Policy Builder

The Policy Builder is the web-based client interface for the configuration of policies to the Cisco Policy Suite. Initial accounts are created during the software installation with the default CPS install username as **`qns-svn`** and password as **`cisco123`**.

URL to Access Policy Builder Interface:

- For HA: https://<*lbvip01*>:7443/pb

The Policy Builder provides a PAM based and SVN based authentication mechanism to support the authentication of Linux user credentials. The `disablePamAuthentication` flag is used to enable or disable user login and to perform PAM based authentication.

The following tables describes the user roles and credentials supported:

Table 1: User Roles and Authentication Mechanism

Linux access	SVN access	User Access to Policy Builder	User Roles	Authentication Mechanism
Read/Write	Not an SVN user	Yes	Read only	PAM (Linux Systems) (set disablePamAuthentication = false)
Read only	Not an SVN user	Yes	Read only	PAM (Linux Systems) (set disablePamAuthentication = false)
Read/Write	Read/Write	Yes	Admin	PAM (Linux Systems) (set disablePamAuthentication = false)

Linux access	SVN access	User Access to Policy Builder	User Roles	Authentication Mechanism
Read/Write	Read only	Yes	Read only	PAM (Linux Systems) (set disablePamAuthentication = false)
Read only	Read/Write	Yes	Admin	PAM (Linux Systems) (set disablePamAuthentication = false)
Read only	Read only	Yes	Read only	PAM (Linux Systems) (set disablePamAuthentication = false)
Not a Linux user	Read only	Yes	Read only	SVN (set disablePamAuthentication = true)
Not a Linux user	Read/Write	Yes	Admin	SVN (set disablePamAuthentication = true)
Not a Linux user	Not an SVN user	No	Invalid username or password error	PAM/SVN

CPS enables users to be aware of its current privileges while accessing Policy Builder as described below:

- If a user has read-write privilege then ADMIN is displayed adjacent to user name in the GUI.
- If a user has read-only privilege then READONLY is displayed adjacent to user name in the GUI.

The hostname is displayed in the login dialog box and system banner to differentiate between open windows while performing any operation of the CPS system. It indicates which system is being modified and prevents any errors or misconfigurations.

The hostname is displayed when the parameter `-Dhostname=lab` is configured in pb/qns.conf files. If it is not configured in the qns.conf file, it is displayed as a result of the command "hostname" on the server.

The hostname is displayed in the login panel only when the following argument is set to true:

```
-DshowSitenameLogin
```

Enable TACACS+ authentication for Policy Builder by enabling PAM authentication (set -DdisablePAMAuthentication to false) and enabling TACACS+ along with `tacacs_on_ui` flag set to true in `Configuration.csv` file.

Enabling Logout Option

To enable the logout option in Policy Builder, the following parameter must be configured in `/etc/broadhop/pb/pb.conf` file.

- –DlogoutLinkVisibility

To view the **Logout** link on Policy Builder banner, set the parameter to true value.

To support backward compatibility, `-DlogoutLinkVisibility` flag is not present in `pb.conf` by default. If flag is not present, then the value is considered as false.

When the parameter is configured or updated, `restartall.sh` is required.

Caution Executing `restartall.sh` will cause messages to be dropped.

Policy Builder Field Value Validation

The Policy Builder uses the eCore framework to configure UI fields and their data types. The Policy Builder validation is triggered when a field is updated. The validation depends on the data type and valid value that you have defined for the field.

When you start the Policy Builder, the last recorded valid value defined in the eCore is set as the default value. If the value is not set in the eCore, the valid value is taken based on the eCore data type. For numeric data types, the Policy Builder displays 0 as a default value. For string data types, the Policy Builder displays null (empty) as a default value.

The Policy Builder validates the value that is configured in the field.

When you enter a value in the field and the value passes validation, the newly entered value is recorded as the last valid value. If the validation fails, the last recorded valid value is reverted.