CHAPTER 1

Overview

You can operate your Cisco 4700 Series Application Control Engine (ACE) appliance in a single context or in multiple contexts. Multiple contexts use virtualization to partition your ACE into multiple virtual devices or contexts. Each context contains its own set of policies, interfaces, resources, and administrators. This feature provides you with the tools to more closely and efficiently manage the system resources and users of the ACE, and the services you provide to your customers.

By default, your ACE provides an Admin context and five user contexts, which allows you to use multiple contexts if you choose to configure them. To increase the number of user contexts up to a maximum of 20, you must obtain a separate license from Cisco Systems. For details about licensing, see the *Cisco 4700 Series Application Control Engine Appliance Administration Guide*.

This chapter provides an overview of the basic concepts involved with virtualization. Virtualization consists of the following functional areas:

- Contexts
- Domains
- Role-Based Access Control
- Resource Classes

Contexts

The virtualized environment is divided into objects called contexts. Each context behaves like an independent ACE appliance with its own policies, interfaces, domains, server farms, real servers, and administrators. Each context also has its own management VLAN that you can access using Telnet or Secure Shell (SSH).

As the global administrator (Admin), you can configure and manage all contexts through the Admin context, which contains the basic settings for each virtual device or context. When you log in to the ACE using the console or Telnet, you are authenticated in the Admin context.

The Admin context is similar to other contexts. The difference is that when you log in to the Admin context (for example, using SSH), you have full system administrator access to the entire ACE and all contexts and objects within it. The Admin context provides access to network-wide resources, for example, a syslog server or context configuration server. All global commands for the ACE settings, contexts, resource classes, and so on, are available only in the Admin context.

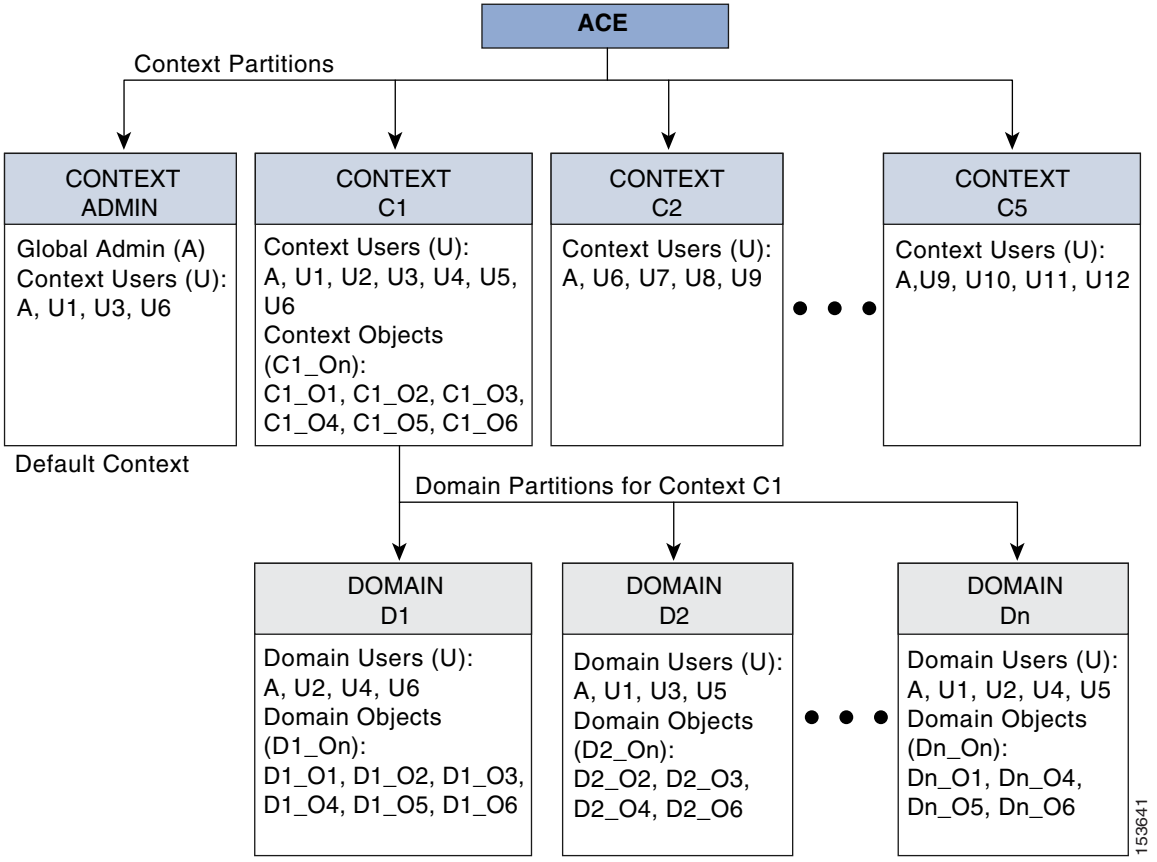
Each context, including the Admin context, has its own configuration file and local user database that are stored in the local disk partition on the flash disk or that can be downloaded from a File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), or HTTP(S) server. The startup-config for each context is stored as the startup configuration file on the flash disk.

In the Admin context, use the **changeto** command in Exec mode or the **do changeto** command in configuration modes to move between contexts. Only users authenticated in the Admin context can use the **changeto** command.

For information about configuring a context, see Chapter 2, Configuring Virtualization.

Figure 1-1 shows how you can use virtualization to create partitions that enable the ACE to function as multiple virtual devices.

Figure 1-1 ACE Virtualization Chart



Each context that you create represents a virtual device. You can partition each context into domains for managing access to context resources. Table 1-1 describes the various components of Figure 1-1.

Table 1-1 ACE Virtualization Elements

Element	Description
Context (**C*n***)	You can configure a single ACE to behave as multiple virtual devices by creating partitions called *contexts*. Each context functions as an independent device with its own set of users, objects, and allocated resources. By default, the ACE comes preconfigured with an Admin context and five configurable user contexts. To upgrade to a maximum of 20 user contexts, you must purchase a separate license from Cisco Systems. For more information about contexts, see the "Contexts" section.
Domain (**D*n***)	You can divide each context into multiple partitions called *domains*, which allow you to manage user access to the objects within a context. When you create a domain, you form an association between a select group of context users and a select group of context objects. For more information about domains, see the "Domains" section.
User (**A**, **U*n***)	The ACE is preconfigured with a default global system administrator that provides access to all ACE functionality and allows you to create additional users. Any user that you create while you are in Admin context, by default, will have access to all resources in the ACE. Any user that you create while you are in a user-defined context will have access only to the resources within that context. You assign each user a role, which determines the commands and resources that are available to that user. For more information about users and user roles, see Chapter 2, Configuring Virtualization.

Table 1-1 ACE Virtualization Elements (continued)

Element	Description
Object (**C*n*_O*n***, **D*n*_O*n***)	The following objects are user-configurable items: • Access lists • Defined interfaces • Policy maps • Health probes • Real servers • Server farms • Scripts • Sticky groups The objects that you create are specific to the context that you are in while creating the object. If the context is partitioned into multiple domains, you allocate objects within each domain.

Domains

For management purposes, contexts are divided into objects called *domains* and each domain is fully contained within a context. A domain provides a namespace in which a user operates and each user is associated with at least one domain. The role assigned to a user determines the operations that a user can perform on the objects in a domain and the command set available to that user. When you create a context, the ACE automatically creates a default domain for that context.

The global admin or context administrators can create additional domains. A domain name must be unique within the context with which it is associated.

You can add any object that you can create (for example, a server farm, a real server, a probe, a VLAN, and so on) to a domain, and you can add an object to multiple domains. If you add an object that has other objects associated with it (for example, a server farm configured with real servers) to a domain, the associated objects do not automatically become part of the domain. You must add each object individually. When you create an object, the ACE automatically adds it to your domain.

Note A domain does not restrict the context configuration that you can display using the **show running-config** command. However, a domain does restrict a user's access to configurable objects in the ACE. You can further restrict the operations that a user can perform on those configurable objects by assigning a role to a user. For information about user roles, see the "Role-Based Access Control" section.

For information about configuring a domain, see Chapter 2, Configuring Virtualization.

Role-Based Access Control

The ACE provides role-based access control (RBAC), which is a mechanism that determines the commands and resources available to each user. A role defines a set of permissions that allow you to access the objects and resources in a context and the actions that you can perform on them. The global or context administrator assigns roles to users based on their network function and the resources to which you want them to have access.

The ACE provides the following predefined roles that you cannot delete or modify:

- Admin—If created in the Admin context, has complete access to, and control over, all contexts, domains, roles, users, resources, and objects in the entire ACE. If created in a user context, this role gives a user complete access to and control over all the objects in that context. A context administrator can create, configure, and modify any object in that context, including policies, roles, domains, server farms, real servers, and so on.
- Network Admin—Complete access to and control over the following features:
  - Interfaces
  - Routing
  - Connection parameters
  - Network Address Translation (NAT)
  - VIPs
  - Copy configurations
  - **changeto** command

- Network-Monitor—Access to all **show** commands and the **changeto** command only. If you do not explicitly assign a role to a user with the **username** command, this is the default role.
- Security-Admin—Complete access to and control over the following security-related features within a context:
  - Access control lists (ACLs)
  - Application inspection
  - Connection parameters
  - Interfaces
  - Authentication, authorization, and accounting (AAA)
  - NAT
  - Copy configurations
  - **changeto** command
- Server-Appln-Maintenance—Complete access to and control over the following features:
  - Real servers
  - Server farms
  - Load balancing
  - Copy configurations
  - **changeto** command
- Server-Maintenance—Real server maintenance, monitoring, and debugging for the following features:
  - Real servers—Modify permission
  - Server farms—Debug permission
  - VIPs—Debug permission
  - Probes—Debug permission
  - Load balancing—Debug permission
  - **changeto** command—Create permission

- SLB-Admin—Complete access to and control over the following ACE features within a context:
  - Real servers
  - Server farms
  - VIPs
  - Probes
  - Load balancing (Layer 3/4 and Layer 7)
  - NAT
  - Interfaces
  - Copy configurations
  - **changeto** command
- SSL-Admin—Administrator for all Secure Sockets Layer (SSL) features:
  - SSL—Create permission
  - Public key infrastructure (PKI)—Create permission
  - Interfaces—Modify permission
  - Copy configurations—Create permission
  - **changeto** command—Create permission

In addition to these predefined roles, Admins in any context can define new roles. For more information, see Chapter 2, Configuring Virtualization.

Resource Classes

Resource classes allow you to manage context access to ACE resources, such as concurrent connections or bandwidth rate. The ACE is preconfigured with a default resource class that it applies to the Admin context and any user context upon creation. The default resource class is configured to allow a context to operate within a range that can vary from no resource access (0 percent) to complete resource access (100 percent).

When you use the default resource class with multiple contexts, you run the risk of oversubscribing ACE resources because the ACE permits all contexts to have full access to all of the resources on a first-come, first-served basis. When a resource is utilized to its maximum limit, the ACE denies additional requests made by any context for that resource.

To avoid oversubscribing resources and to help guarantee access to a resource by any context, the ACE allows you to create customized resource classes that you associate with one or more contexts. A context becomes a *member* of the resource class when you make the association. Creating a resource class allows you to set limits on the minimum and maximum amounts of each ACE resource that a member context is entitled to use. You define the minimum and maximum values as a percentage of the whole. For example, you can create a resource class that allows its member contexts access to no less that 25 percent of the total number of SSL connections that the ACE supports.

You can limit and manage the allocation of the following ACE resources:

- Application acceleration connections
- ACL memory
- Buffers for syslog messages and TCP out-of-order (OOO) segments
- Concurrent connections (through-the-ACE traffic)
- Management connections (to-the-ACE traffic)
- Proxy connections
- Set resource limit as a rate (number per second)
- Regular expression (regexp) memory
- SSL connections
- Sticky entries
- Static or dynamic network address translations (Xlates)

By default, when you create a context, the ACE associates the context with the default resource class. The default resource class provides resources of a minimum of 0 and a maximum of unlimited for all resources except sticky entries. For stickiness to work properly, you must explicitly configure a minimum resource limit for sticky entries by using the **limit-resource** command.

For more information about configuring and limiting resources, see Chapter 2, Configuring Virtualization. For more information about stickiness, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide*.