**Chapter 12: Device Administration**

Add a note hereSOA Appliances have the potential to be accessed by a number of groups with varied roles in an enterprise; networking, security, and application development groups can find uses for the appliance functionality. Their cross-departmental appeal is an asset to the organization but can be a headache for a device administrator, who must now define who has access to the appliance and at what granularity. The enforcement of this access will depend heavily on corporate governance policies concerning user access, including passwords and permissions. Fortunately, the architecture is engineered so that every user can have access to an appropriate subset of the appliance’s configuration interfaces and features. This chapter provides a device administrator the knowledge necessary to create and manage domains, users, and groups on development, test, and production devices. It explains the built-in user groups and the creation of new groups with custom permission profiles. It also demonstrates administration of device access control using off-device authentication mechanisms for role-based management using existing identity management tools.

**Add a note here****Application Domains**

Add a note hereDataPower appliances are divided into logical partitions called application domains. Think of them as different virtual machines running in a server for application isolation, or (at a high level) logical partitions (LPAR) defined in a mainframe/iSeries® environment. Domains provide the capability to create services, upload files, and perform other configuration tasks without interfering with the services in any of the other domains. They can be used as collaborative sandboxes, project development spaces, or even as test platforms for single users. Though the precise granularity of a domain and its users is entirely flexible, the most common scenario is to create domains for an application development group or project. When used for an entire development group, domains enable sharing of files and reusable policies such as access control definitions or matching rules. Services can be defined collaboratively, with a security-focused developer creating Web services security configurations while mainframe developers create maps between non-XML and XML formats in a transformation service. When used for a specific development project, a domain can act as a *unit for environment promotion*. When the development work is done, active services from within that domain can be copied onto a QA device for testing. These updates won’t affect other services that are already running in other domains, with the possible exception of network-level port conflicts, which we discuss later in this chapter.

Add a note hereUpon initialization of a new device, there is only one predefined domain: default. The default domain is special in that it contains the network and device management operations, such as user creation and Ethernet interface configuration. By isolating these functions from everyday usage, the default domain protects vital resources from normal developers. A device administrator will spend much of his time in the default domain, configuring connections to DNS and time servers, creating additional domains for new projects, and allocating user permissions."Administration Versus Development Domains"

**Administration Versus Development Domains**

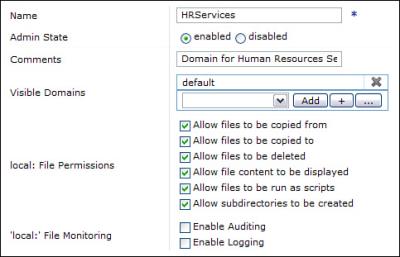
Add a note hereChanges to the default domain should *always* be restricted to device configuration only; separate application domains should be used for any application-specific proxies or configurations.

Add a note herePlease note that although domains are logically separate from a service and file perspective, they do share the same pool of resources (memory, CPU) and network-level connections. On each device, there can be only one listener configured per IP:port combination. Thus, if a service or Front Side Handler in one domain is listening on port 80, no listener in another domain can be configured on the same IP address at port 80. If the architecture requires services to be listening on standard HTTP and HTTPS ports (80 and 443), then secondary IP addresses can be configured for use in different domains; that is, an Ethernet interface can have two IP addresses, each used by listeners in a specific domain. A list of currently active ports can be found by navigating to Status→IP-Network→TCP Port Status from the left-side menu.

Add a note hereAlternatively, a simple pass-through service can be defined to listen on port 80 and route all traffic to the appropriate DataPower service on another port—even if that service is running in a different domain. If a service backend is defined using the localhost IP, 127.0.0.1, DataPower-defined services can be called without exiting to the network stack and running into firewall complications. This works on port 443 for terminating SSL connections as well, but keep in mind that only a single certificate can be presented on any IP:port combination due to the nature of SSL.

**Add a note here****Creating Domains**

Add a note hereGenerally, creating a domain involves only the single step of providing a name, as shown in Figure 12-1, though there are certainly many configuration options that can be tweaked if necessary.

[](javascript:PopImage('IMG_286','http://images.books24x7.com/bookimages/id_30903/12fig01.jpg','483','311'))  
Add a note hereFigure 12-1: Adding a new application domain.

Add a note hereThe domain name is the only required field, while the default values for the other fields provide a reasonable level of access for a domain that is currently being used for development. In a domain on a QA or production device, some local: file permissions can be removed for additional security. For enterprises that require an audit trail any time an in-use file is modified, auditing or logging can be enabled on the local: directory as well.

Add a note hereThe Visible Domains property provides some level of access to the file system in other domains. The default domain, which includes the store: directory containing commonly used, DataPower-provided files, is always included in a new domain’s list, enabling services in this domain to have access to files in the store: directory. Adding other visible domains to this list will create additional links from the File Management screen within the domain being created. If a hypothetical CoreServices domain is added to the list when creating the HRServices domain, then the File Management screen within HRServices will include a CoreServices: directory for design-time copying of shared files. This connectivity is available *only* during the design phase; with the exception of the store: directory, files within these visible domains cannot be referenced during runtime execution.

**Add a note here****Remote Domain Configuration**

Add a note hereDomain configurations, including the services and objects that have been created within, are generally stored on the local file system (in each domain’s config: directory) and loaded at device startup. The Add Domain screen’s Configuration tab contains an interesting alternative; domain configurations can be loaded from remote servers at startup. In Figure 12-2, the HRServices domain will be retrieved from a configuration server.

[](javascript:PopImage('IMG_287','http://images.books24x7.com/bookimages/id_30903/12fig02.jpg','436','158'))  
Add a note hereFigure 12-2: Defining a remote domain configuration file.

Add a note hereBy choosing Import as the configuration mode and providing the URL of a remote configuration that has been exported from the device, domains can be dynamically generated at startup and shared between multiple devices. Every device in a cluster can point at a single golden exported copy of a domain, perhaps stored in a version control system; updates to that domain on every device would simply require a domain restart or device reboot after modifying the configuration file. Another advantage of this approach is that the limited available space on the DataPower flash memory is conserved. This is a requirement of certain high-security environments, which do not allow configuration files to be stored on the file system of DMZ devices (even highly secure ones).

Add a note hereIf you choose this configuration method, there are a few things to keep in mind:

* Add a note hereThe domain should be hosted on a highly available server or cluster of servers. If the server is down and the configuration cannot be retrieved, the domain will be down upon device startup. An alert is generated to warn you of this status.
* Add a note hereThe server should use mutually authenticated SSL for its communication; this prevents substitution attacks and protects the configuration file in-transit.
* Add a note hereThe export should be backed up and stored in a source code repository to make sure that hardware failures and other server problems do not permanently destroy a configuration.
* Add a note hereKeys, certificates, and other files stored in the cert: directory need to be uploaded separately from the configuration, as these files will not be included in the export for security purposes.

Add a note hereCustomizing an imported configuration using deployment policies and IP rewriting are discussed in [Chapter 15](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2720#2720), [“Build and Deploy Techniques.”](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2720#2720) Domain configurations can also be “pushed” to a device using ITCAM[[1](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=469752441&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ftn.ch12fn01)] System Edition for DataPower. In this case, the Configuration Mode field would retain its default value of local.

**Add a note here****Managing Domains**

Add a note hereManagement of domains for environment promotion is discussed in [Chapter 15](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2720#2720); however, for the purposes of this chapter, you need to know how a device administrator performs basic management tasks, including domain exports, checkpoints, and comparisons.

**Configuration Exports**

Add a note hereTo back up a full domain for insertion into a code repository, simply click the Export Configuration link from the Control Panel. Choosing “Create a backup of one or more application domains” allows you to select any domains subject to continuous development and download them to a local machine. The end result is a .zip file that can be backed up or saved on a local machine for manual import on another device. If exported in this manner and imported on another device, these domains will be created as new.

Add a note hereTo export a configuration for hosting on an HTTP server for remote domain configuration, as discussed earlier in this chapter, choose “Export configuration and files from the current domain,” and then select All Objects. This results in an appropriately formatted .zip file for dynamic configuration imports. This export method also enables the services to be imported into *any* domain, as the domain itself is not included.

**Configuration Checkpoints**

Add a note hereWhile performing day-to-day configuration changes, especially when making modifications that may potentially modify existing behavior, it may be helpful to take a snapshot of the current state without performing a full domain export. To take an on-device snapshot of a domain’s current configuration, a checkpoint can be created. From within the domain in question, select Administration→Configuration→Configuration Checkpoints from the left menu. Provide a unique, descriptive checkpoint name and click Save Checkpoint. In Figure 12-3, we created a checkpoint before making modifications to the current configuration.

[Image from book](javascript:PopImage('IMG_288','http://images.books24x7.com/bookimages/id_30903/12fig03.jpg','432','45'))  
Add a note hereFigure 12-3: Saving a configuration checkpoint.

Add a note hereA saved checkpoint references its name, the time it was taken, and the various operations available, as seen in Figure 12-4.

[Image from book](javascript:PopImage('IMG_289','http://images.books24x7.com/bookimages/id_30903/12fig04_alt.jpg','531','50'))  
Add a note hereFigure 12-4: Configuration checkpoint options.

Add a note hereChoosing Rollback reverts the domain configuration to the state saved in the checkpoint. All changes made since the timestamp are lost; be sure the checkpoint names are descriptive enough to confirm appropriate use of the checkpoint you have chosen. Choosing Remove deletes the configuration checkpoint, freeing room for new ones to be created. The number of checkpoints allowed per domain is configurable from within the Application Domain object.

Add a note hereOn-device checkpoints are not a replacement for a strong IT backup policy, which should incorporate the backup of your DataPower configurations. There is limited on-board file system space, so don’t get carried away with hundreds of checkpoints. If the flash file system or the device itself happens to experience a hardware failure, these checkpoints would not be retrievable. However, this is a helpful feature for taking quick snapshots of configurations during service development.

**Configuration Comparisons**

Add a note hereBefore performing a rollback, saving a currently running domain to the persisted file system, or importing a slightly different version of a domain, it may be helpful to compare two potential configurations. Revealing their differences can help you make choices about saving, rolling back, or importing a domain.

Add a note hereTo compare any two domain configurations, including checkpoints, external config files, and the currently running configuration, navigate to Administration→Configuration→Compare Configuration, or click Compare next to a specific checkpoint. This opens the Configuration Comparison screen seen in Figure 12-5.

[](javascript:PopImage('IMG_290','http://images.books24x7.com/bookimages/id_30903/12fig05_alt.jpg','652','245'))  
Add a note hereFigure 12-5: Comparing a saved checkpoint to the currently running configuration.

Add a note hereConfiguration comparison can determine the modified objects between any two of the following:

* Add a note here**Persisted Configuration—** The configuration currently saved to the file system for this domain.
* Add a note here**Running Configuration—** The configuration currently running, including any changes made since the last time the configuration was persisted using the Save Config link.
* Add a note here**XML Configuration—** A set of objects (or even all objects from a domain) that have been exported as an XML file.
* Add a note here**Export ZIP Bundle—** A domain or set of objects that has been exported as a zip file.
* Add a note here**Backup ZIP Bundle—** A set of domains that have been exported as a single zip file.
* Add a note here**Checkpoint—** A checkpoint that has been saved for the current domain.

Add a note hereChoose a checkpoint or configuration on each side and select whether to compare every object, or only the changed objects. In domains with a large number of services, you should definitely choose View Changed Objects Only to view a more manageable comparison.

Add a note here[[1](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=469752441&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ch12fn01)]The IBM Tivoli Composite Application Manager is an external software application that provides device management capabilities; see [Chapter 15](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2720#2720) for more details.

**Users**

Add a note hereUsers of DataPower SOA Appliances fall into several categories, from network admins to production support personnel to application developers. Each of these distinct users requires access to different portions of the device, with varying levels of permission to create, read, modify, and delete configuration objects. Keep in mind that these users of the DataPower administration interfaces are very different from application users who might want to execute a service configured on the device.

Add a note hereThere is only one user predefined on the appliance at initialization: the admin user. This super user has full control over everything on the device—essentially unlimited permissions. Think of this account as the root user in a \*nix system, or an Administrator with full access rights on a Windows server. Initial login using this username is discussed in [Chapter 2](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=145#145), [“DataPower Quick Tour and Setup,”](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=145#145) but this point cannot be stressed enough: Don’t lose the admin password![[2](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=973474286&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ftn.ch12fn02)] There is no password reset for the admin user, and if you lose it, the entire device will need to be shipped to DataPower HQ for a re-initialization.

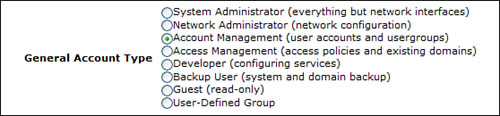
Add a note hereAfter you have performed initial setup steps and logged on to the DataPower WebGUI, you can begin creating users that will administer, develop, or view configurations within that device. User definitions can be defined locally, on the device, or remotely, within an external authentication server. External users will be discussed in the “[Role-Based Management](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2141#2141)” section later in this chapter; the following sections apply to local users defined within the appliance."Admin and Auditing"

**Admin and Auditing**

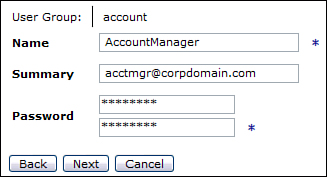
Add a note hereAfter initializing a device as the admin user, you should immediately create new users for performing all subsequent administrative tasks. The time and date of any changes saved to the default domain are recorded in a locked audit log. The audit log is available from Status→View Status→Audit Log in the left-side menu. By distributing varied administrative accounts and enforcing login using only those credentials (and not the admin account), you can provide an audit record and change tracking for any modifications made by these users.

**Add a note here****Creating Users**

Add a note hereTo create a new user, browse to Administration→Access→New User Account in the left-side menu. The first question asked is whether the user should be restricted to a particular domain. If so, the user is created with a custom-defined group (set of permissions) called developer\_*DomainName*. The user is given full access to define services and other objects within the selected domain, and no access to any other domains (including default). While simply restricting the user to a domain may be sufficient for simple implementations or on a generic development appliance, many users require higher-level or custom access rights. If the user is *not* restricted to a particular domain, then they must be placed within a user group to fully define their level of access. User groups provide a mapping between a set of users and the administration permissions provided to them, and can be thought of as administrative roles. Figure 12-6 shows the options for built-in user groups unrestricted by domain.

[](javascript:PopImage('IMG_291','http://images.books24x7.com/bookimages/id_30903/12fig06_alt.jpg','558','130'))  
Add a note hereFigure 12-6: Choosing a group for the new user.

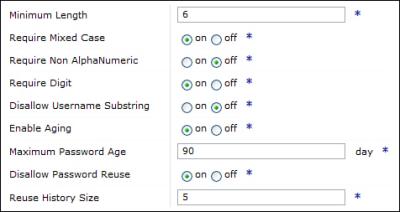
Add a note hereAfter the account type has been defined, you’ll need to provide a username and an initial password. Usernames are not case-sensitive, so feel free to use any combination of uppercase and lowercase letters. During the user’s first login, they’ll have to change their password, so you can pick a random (and secure) string to provide with their username. The default password policy requires them to be at least six characters. You’ll need to type it twice for verification, as shown in Figure 12-7.

  
Add a note hereFigure 12-7: Defining a username/password combination.

Add a note hereThe summary is a good place to provide a comment about the user, including either a description of their role or contact information for that user. A corporate email address in this field would give device administrators a method for contacting the user if necessary.

**Add a note here****Defining a Password Policy**

Add a note hereIf your enterprise requires a specific password policy for local accounts on internal servers, you can define stricter password requirements by navigating to Administration→Access→RBM Settings and clicking on the Password Policy tab. The password options are shown in Figure 12-8.

[](javascript:PopImage('IMG_293','http://images.books24x7.com/bookimages/id_30903/12fig08.jpg','476','253'))  
Add a note hereFigure 12-8: Updating the default password policy for local users.

Add a note hereThough this should be modified to meet your corporate security policy, the most common updates are requiring mixed case and either a digit or a nonalphanumeric character. Enabling aging enables you to define a maximum password age. Disallowing password reuse retains a user-configurable number of old passwords to check for duplication.

Add a note here[[2](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=973474286&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ch12fn02)]We mean it. Do not lose this password. Shipping an appliance to Massachusetts is a lot of trouble. Just remember the password.

**User Groups**

Add a note hereUser groups provide an easy way to combine individual permissions into a more comprehensive access profile. There are both built-in and user-defined groups, allowing for generic roles or fine-grained object access.

**Add a note here****Built-In User Groups**

Add a note hereBuilt-in user groups provide analogs to common roles within an organization so that most users can simply fit in to the category that best describes their job role. Though a brief description is supplied in the WebGUI, how do these map to real-life permissions?

**System Administrator**

Add a note hereThe system administrator can modify everything except network interfaces. The system administrator has these permissions:

* Add a note hereRead-only access to store: directory on file system
* Add a note hereRead-only access to Ethernet interfaces
* Add a note hereFull access to all other objects on the device

**Network Administrator**

Add a note hereThe network administrator can perform the network-level tasks that the system administrator cannot and has these permissions:

* Add a note hereRead-write access to all network-level objects, including Ethernet interfaces, device access points (WebGUI, CLI, SSH, Telnet), DNS and NTP servers, and SNMP configurations
* Add a note hereRead access for all network-related status screens, including routing tables, ARP tables, and server connectivity
* Add a note hereExecute access for network troubleshooting operations such as pings, packet captures, and error reports

**Account Management**

Add a note hereAn account manager can create [users](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2072#2072) and groups on the device and has these permissions:

* Add a note hereLogin access to the WebGUI and SSH/Telnet interfaces
* Add a note hereFull access to create users and user groups, with the exception of privileged users
* Add a note hereExecute access for modifying log levels for Role-Based Management

**Access Management**

Add a note hereAn access manager can set up login authentication to use a remote server and has these permissions:

* Add a note hereRead-write access to all interfaces (WebGUI, XML Management, SSH, Telnet) to modify access policies
* Add a note hereFull access to create users and user groups
* Add a note hereAbility to modify Role-Based Management settings to perform login authentication on a remote authentication server

**Developer**

Add a note hereA developer can create and modify services in all domains and has the following permissions:

* Add a note hereRead-only access to users, domains, network interfaces, and system information
* Add a note hereNo access to system control operations such as device shutdown and firmware upgrades
* Add a note hereFull access to all other objects, allowing configuration of services in any domain

**Backup User**

Add a note hereA backup user can perform operations for configuration management and has the following permissions:

* Add a note hereExecute access to checkpoint operations (save, rollback, remove)
* Add a note hereExecute access to backup operations
* Add a note hereRead access to checkpoint status information

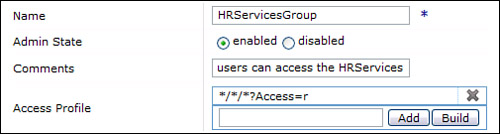
**Guest**

Add a note hereA guest is limited to purely read operations on all objects and has the following permission:

* Add a note hereRead-only access to entire device

**Add a note here****Custom User Groups**

Add a note hereIn situations when there is no built-in user group that contains the exact set of desired permissions, such as users who are limited to an admin-defined selection of domains, a custom user group can be constructed from granular permission statements. These groups can then be reused for all users that require this particular access policy. To create a new user group, browse to Administration→Access→Manage User Groups in the left-side menu and click Add. This opens the configuration page seen in Figure 12-9.

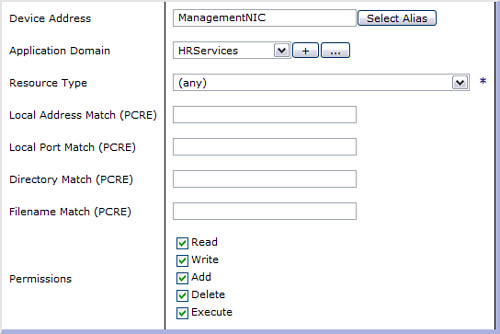
[](javascript:PopImage('IMG_294','http://images.books24x7.com/bookimages/id_30903/12fig09_alt.jpg','505','135'))  
Add a note hereFigure 12-9: Creating a new user group.

Add a note hereA user group consists of a name and an associated access profile. Access policies are built individually and then added to the list to create a profile. Provide a descriptive name so that other administrators provisioning users will not need to look up the specific permissions within a custom group.

**Add a note here****Access Profiles**

Add a note hereAccess profiles consist of a set of access policies that combine to form a level of access appropriate to members of the group. When a user attempts to perform an operation (read, write, add, delete, or execute) on a resource within DataPower, these policies are checked for the permissions that have been applied to that resource. When more than one policy applies, the most specific policy will be in effect. Thus it is possible to define broad, generic policies that apply to large groups of objects, and more specific statements that will override the generic rule for certain resources.

Add a note hereClick Build within a user group, as demonstrated in Figure 12-9, to open the access policy window shown in Figure 12-10.

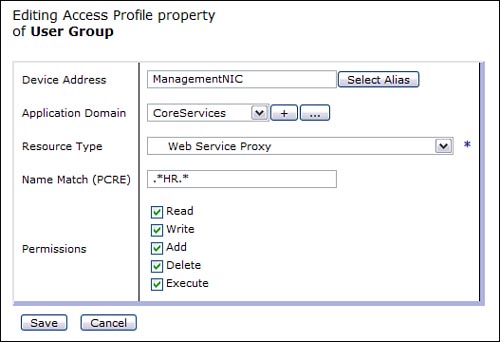
[](javascript:PopImage('IMG_295','http://images.books24x7.com/bookimages/id_30903/12fig10_alt.jpg','544','363'))  
Add a note hereFigure 12-10: Creating an access policy.

Add a note hereAccess policies use a top-down approach to defining a resource. In Figure 12-10, you can see the properties, from Device Address to Permissions; these will compose an access policy statement. The syntax for an access policy is *Address*/[*Domain*](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2493#2493)/*Resource*?Access=*permissions*[&field=*value*].

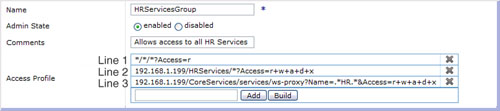
* Add a note here*Address* is the device address on which this resource can be accessed. This can limit a user to logging in for device management via a particular Ethernet interface. It must take the form of a fully specified hostname, an IP address, or a local host alias. For access on all addresses, an asterisk (\*) can be used to designate any.
* Add a note here[*Domain*](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2493#2493) is the application domain containing the resource in question. This must either be the complete name of a domain or an asterisk to represent any domain.
* Add a note here*Resource* is the type of resource being accessed. Choose from the drop-down list any of the resources on the device; they are divided into categories for easy access. Due to the large number of potential options, it is best to define a generic policy giving the default permissions, and then define a few more specific policies to override the defaults in particular cases. Selecting (all resources) will use an asterisk to designate all resources.
* Add a note here*Permissions* are the set of operations that a user can perform on this resource. They include read, write, add, delete, and execute. Any combination of these operations can be added to a single access policy, though they don’t all apply to every resource. For example, any resource in the status category requires only read access, because these are all read-only objects. Similarly, resources that perform operations such as “change-password” or “rollback-chkpoint” only require execute access. Services and their related objects (handlers, processing policies, rules, and actions) can be added, deleted, read, or modified (write).
* Add a note here*Fields* available for matching a specific object will depend on the object type. A File resource may have a Directory field and a Name field on which to match, while a Web Service Proxy object will have only a Name field. Any Field/Value pairs provided here serve to limit the object to which this access profile applies. You can see an example of a Name match in the access profiles depicted in Figure 12-11.
* Add a note here*Values* provided for a given field are PCRE (Perl Compatible Regular Expression) statements that define specific objects or groups of objects. By using wildcards and other PCRE designators, single policy statements can apply to multiple resources.

Add a note hereAfter filling in the appropriate fields in the access policy, click Save to return to the user group definition. Add the new policy to the list, and then build as many new policy statements as are required to complete the user group’s access profile.

Add a note hereEvery new user group has a single access policy defined by default. The generic read-only access policy (\*/\*/\*?Access=r) can be removed or modified to provide more privacy for other domains. Choosing a resource type and providing a filter can add specific access to additional objects. In Figure 12-11, this group is given full permissions to all current and future Web Service Proxies with names containing the string “HR” that are defined in the CoreServices domain.

[](javascript:PopImage('IMG_296','http://images.books24x7.com/bookimages/id_30903/12fig11_alt.jpg','527','360'))  
Add a note hereFigure 12-11: Defining a filter on a subset of domain objects.

Add a note hereIn the full user group definition shown in Figure 12-12, the HRServicesGroup has read-only access to every domain (Line 1), full access to the HRServices domain (Line 2), and access to a subset of services in the CoreServices domain: those with HR in their name (Line 3). Any Web Service Proxies without HR in their names are subjected to the read-only permissions defined in Line 1.

[](javascript:PopImage('IMG_297','http://images.books24x7.com/bookimages/id_30903/12fig12_alt.jpg','808','180'))  
Add a note hereFigure 12-12: Adding multiple access policies to a custom user group.

Add a note hereHere are a few example policies, from the Network Administrator group:

* Add a note here\*/\*/access/change-password?Access=x allows the user to change their own password.
* Add a note here\*/\*/config/rollback-chkpoint?Access=x allows the user to roll back to an earlier configuration checkpoint.
* Add a note here\*/\*/debug/ping?Access=x allows the user to ping remote hosts to test network connectivity.
* Add a note here\*/\*/mgmt/web-mgmt?Access=rw allows the user to view and modify the configuration of the WebGUI management interface.
* Add a note here\*/\*/network/interface?Access=rwad allows modification of the Ethernet interface configurations.

Add a note hereYou can get additional samples from any of the built-in user groups defined on the device,[[3](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=746733105&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ftn.ch12fn03)] Custom roles often need a combination of rights; if an account manager also needs some development rights, you can combine the access policies (by hand) into an appropriate custom role.

Add a note here[[3](http://www.books24x7.com/assetviewer.aspx?bookid=30903&chunkid=746733105&noteMenuToggle=0&hitSectionMenuToggle=0&leftMenuState=1" \l "ch12fn03)]Note that these built-in groups are not available from the Manage User Groups screen until at least one user has been assigned to the group. They are only added to the list as necessary to fit your pool of users.

**Role-Based Management**

Add a note hereFor devices that will only ever be accessed by a few users, especially in nondevelopment environments that should be locked down to QA/production support teams, the local users and groups defined earlier in this chapter should suffice for user authentication. However, most enterprises have a corporate directory stored in LDAP or another registry and might want to leverage the accounts there, rather than create yet another account ID and password for their employees. This is especially true for development groups with large numbers of potential users of the appliance. In these cases, DataPower’s Role-Based Management (RBM) policy can use external authentication servers for user login. Browse to Administration→Access→RBM Settings to modify the login method.

Add a note hereAs you read the instructions that follow, keep in mind that any changes made on this screen will take effect immediately upon clicking Apply in the RBM policy. Make sure that your authentication server is available and appropriately configured, or you could be locked out of the WebGUI and need to rely on a fallback user or CLI override. If you have access to the physical device, you can use the following steps to double-check your changes:

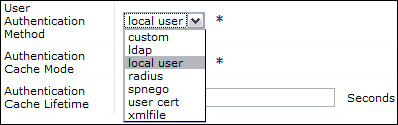
1. Add a note herePerform your RBM modifications.
2. Add a note hereClick Apply to enforce the changes, but do not click Save Config.
3. Add a note hereVerify appropriate access for two users, one affected by the change and one not affected.
4. Add a note hereIf access is not as expected, physically restart the device; the configuration (including RBM changes) will be rolled back to the last saved state."Reliance on External Servers"

**Reliance on External Servers**

Add a note hereAs with all configuration options that require connectivity to external servers, RBM modifications should be approached with care. High availability and guaranteed connectivity for authentication servers is especially important in this case, as a misconfiguration or network failure could impact your ability to log in to the appliance.

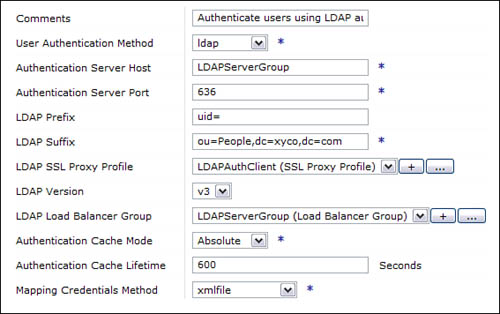
**Add a note here****Authenticating Users**

Add a note hereThe first step of any login process is to authenticate the user. For local users, the RBM settings need not be modified in any way, though you can strengthen the on-device password policies if necessary, as we demonstrated in [Figure 12-8](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2088#2088). If there is a requirement to check usernames and passwords against a remote authentication server such as LDAP, rather than the local device, you can choose from one of the supported server types shown in Figure 12-13.

  
Add a note hereFigure 12-13: Selecting a remote authentication method for RBM.

Add a note hereWhen using LDAP or RADIUS, the user must still log in from the main DataPower login page. The username and password provided are extracted and sent to the authentication server. Depending on the server type, you’ll need to provide various other fields. LDAP servers require a host:port combination, as well as a prefix and suffix to append to the username to create a valid Distinguished Name (DN); DataPower performs a full DN search on the registry using the newly created string. Authentication against a RADIUS server requires a predefined server definition in Administration→Access→RADIUS Settings. SPNEGO access will require a Kerberos keytab for decrypting the incoming ticket. Users can also be defined in an XML file that can be hosted on a remote server and modified by security personnel.

Add a note hereAuthentication using LDAP is the most common use case, and requires some additional parameters as described previously. In Figure 12-14, the authentication call uses a load balanced group of LDAP servers for high availability purposes. The full DN is built from the prefix, suffix, and login username. For example, logging in as AUser attempts to authenticate using the following DN: uid=AUser,ou=People,dc=xyco,dc=com. It also is transmitted using SSL, to ensure transport encryption of the submitted username/password.

[](javascript:PopImage('IMG_299','http://images.books24x7.com/bookimages/id_30903/12fig14_alt.jpg','567','356'))  
Add a note hereFigure 12-14: RBM configuration using an LDAP authentication server.

Add a note hereIf the DataPower device cannot contact the remote authentication server due to network or other connectivity problems, all appliance (administrative) login attempts will fail. In order to prevent device lockout, set Local Login As Fallback to either all [users](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2072#2072) or specific [users](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2072#2072). If set for specific [users](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2072#2072), define the subset of local [users](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2072#2072) that are allowed to log in using their local credentials, as shown in Figure 12-15. This backup authentication mechanism will only be invoked in the case of network or infrastructure failure preventing access to the primary authentication mechanism.

[](javascript:PopImage('IMG_300','http://images.books24x7.com/bookimages/id_30903/12fig15.jpg','473','98'))  
Add a note hereFigure 12-15: Defining specific users for local fallback access.

Add a note hereIn addition, for complete administrative access to the device, even when successfully connecting to your backend authentication server, the registry in question must have an entry for the admin user. In most cases, this requires you to add such a user to your LDAP/RADIUS credential store. Make sure to complete this addition before updating your RBM settings.

**Add a note here****Mapping User Credentials**

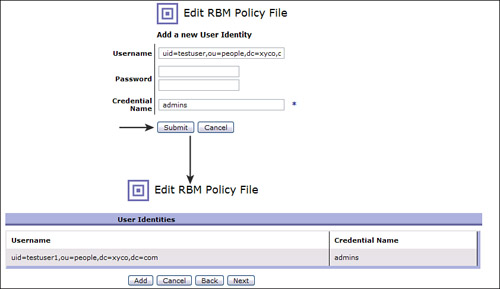
Add a note hereAn authenticated user must then be mapped to a credential; the credential is then used to determine the authorization status of the user for any particular resource. When employing local access management, the user is assigned to a particular user group, which assigns an access profile that determines authorization. Authentication done using an on-board XML file also allows mapping between an input credential (username) and an output credential (group). When the user is authenticated off-device, you can define both the input→output credential mapping and the mapping of output credential to access profile.

Add a note hereThe easiest way to perform these mappings and generate an appropriate access profile is to use the xmlfile method of credential mapping. This requires a Mapping RBM Policy URL, which can reference a local or remote XML file that follows the proper format. If you create a new RBM Policy File from the RBM Settings page (click + next to the Mapping RBM Policy URL), a wizard takes you through each step in the process. Let’s walk through each step in the wizard; at the end, you’ll have a new XML file that maps user credentials to a set of access policies.

Add a note hereFirst, you can define a default credential given to unauthenticated users. If you want everyone with network access to the device to have read-only permission to login and view configurations, you could define a “[guest](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2113#2113)” default credential as demonstrated in Figure 12-16. Leaving this blank will simply reject any user that doesn’t have an authenticated user credential. Unless you have an overwhelming need for allowing *all* potential users some level of access, the default credential should be left blank.

[](javascript:PopImage('IMG_301','http://images.books24x7.com/bookimages/id_30903/12fig16.jpg','446','91'))  
Add a note hereFigure 12-16: Defining a guest credential for unauthenticated access.

Add a note hereIn the next wizard screen, map from the username to a credential name. Define your input credential and the output credential that will be created. If your authentication method was LDAP, your username (input credential) will no longer be just a username; it will be the distinguished name that was sent to the authentication server (see Figure 12-17).

[](javascript:PopImage('IMG_302','http://images.books24x7.com/bookimages/id_30903/12fig17_alt.jpg','842','487'))  
Add a note hereFigure 12-17: Mapping a user identity to an output credential.

Add a note hereAdding the appropriate Username and Credential Name entry in the wizard, as shown in Figure 12-17, will produce a mapping statement in the underlying XML file, as shown in Listing 12-1.

Add a note hereListing 12-1: Mapping an LDAP DN to an output credential

Add a note here<!-- Map credentials to different credentials. -->

<aaa:MapCredentials>

<aaa:InputCredential>

uid=testuser1,ou=people,dc=hr,dc=xyco,dc=com

</aaa:InputCredential>

<aaa:OutputCredential>admins</aaa:OutputCredential>

</aaa:MapCredentials>

Add a note hereThese credential mappings also allow wildcards, so that uid=.\*,ou=people,dc=hr,dc=xyco,dc=com allows you to map any authenticated credential to an output credential.

Add a note hereThe next step in the wizard allows you to map from this output credential to an access profile consisting of one or more permissions statements. In the example in Figure 12-18, the admins are being mapped to a powerful access policy that gives all permissions on all device objects.

[](javascript:PopImage('IMG_303','http://images.books24x7.com/bookimages/id_30903/12fig18.jpg','451','176'))  
Add a note hereFigure 12-18: Creating an access profile for a mapped credential.

Add a note hereDefining a set of these access profiles for each mapped credential creates a list similar to that in Figure 12-19. When the full list of possible credentials is mapped, the RBM file is complete. Finish the wizard by defining a filename and saving it to the local filesystem. The file can then be exported (if desired), and referenced remotely.

[](javascript:PopImage('IMG_304','http://images.books24x7.com/bookimages/id_30903/12fig19_alt.jpg','803','137'))  
Add a note hereFigure 12-19: Mapping incoming credentials to specific access rights.

**Add a note here****Using Access Profiles**

Add a note hereJust as they must be defined from their individual access policies for custom user groups, Access Profiles must also be defined on each RBM Output Credential. Figure 12-19 demonstrates an example of this mapping. Access profile basics, more examples, and a complete dissection of the profile string syntax are discussed in detail in their own section earlier in this chapter.

Add a note hereConfiguring each step in the wizard produces an XML file that can be named and saved to the local: directory. When used as the RBM Policy Mapping file, this enforces the appropriate permissions for users authenticated using off-device credential stores.

**User Authentication Caching**

Add a note hereWhen a user is authenticated using a credential store, either on the device or off, that authentication decision is cached by default. This caching means that the authentication server does not need to be accessed on every request, to reduce network operations for a lookup that will likely change very infrequently. The caching mode and duration can be defined from the RBM Settings page, as shown in Figure 12-20.

[Image from book](javascript:PopImage('IMG_305','http://images.books24x7.com/bookimages/id_30903/12fig20.jpg','463','51'))  
Add a note hereFigure 12-20: Defining a caching policy for authenticated users.

Add a note hereA setting of Absolute always uses the cache lifetime defined below it; the default is 600 seconds, or 10 minutes. By choosing a different cache mode, this caching can be disabled or tied to a TTL (time-to-live) provided by the authentication server, if such a parameter was included in the authentication response. Maximum uses the higher of the two values, between the user-defined cache lifetime and the TTL, while minimum uses the lower of the two values.

Add a note hereIf your enterprise has uncharacteristic changes performed on its authentication servers, so that a large number of authentication decisions need to be invalidated and retrieved from scratch, you can require this revalidation from the WebGUI. Along the top of the RBM Settings page are the helpful links shown in Figure 12-21, including Flush RBM Cache.

[Image from book](javascript:PopImage('IMG_306','http://images.books24x7.com/bookimages/id_30903/12fig21_alt.jpg','802','59'))  
Add a note hereFigure 12-21: Flushing the authentication cache from the RBM Settings page.

Add a note hereBy flushing the cache, each new request checks against the authentication store. The current cache settings are still in effect, so all responses after the flush will be cached as desired.

**Access to the Command Line Interface**

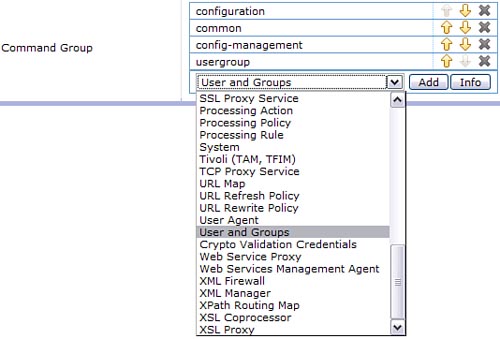
Add a note hereManagement of access to the Command Line Interface (CLI) is a bit more flexible—it can use the same rules that have been defined for WebGUI access, or it can be separately defined in a CLI-specific manner. To define all CLI access to use the exact same policies as the WebGUI, navigate to Administration→Access→RBM Settings. Set the Enforce RBM on CLI option to on. This will override all other CLI permissions settings discussed in the following sections. Because groups of CLI commands can’t be delegated to an external registry, administrative access to the CLI is often enforced using the following methods instead.

Add a note hereIf the WebGUI login settings are not used for the CLI as well, then there are two steps for determining whether a user will have access to a particular operation within a domain. First, a user group is given access to specific operation categories from the User Group configuration page, as described in the [next section](http://www.books24x7.com/assetviewer.aspx?bkid=30903&destid=2190#2190). Then our discussion will shift to the following step, where a user is given the rights to a domain from the Application Domain configuration page.

**Add a note here****Command Groups**

Add a note hereA command group is a group of associated commands that fit within a given category. For example, the File Management command group gives permission for operations on both the local file system and remote file servers, by allowing the definition of NFS mount objects. These groups provide a method for provisioning commands without having to define every specific command that is allowed; however, the granularity should still be small enough that no unnecessary permissions need to be granted.

Add a note hereTo give a user group access to a command group, navigate to Administration→Access→Manage User Groups from the left-side menu, and then choose a user group on which to apply CLI permissions. Select the CLI Command Groups tab to expose the drop-down list of potential command groups that can be set for this user group, as seen in Figure 12-22. In this example, the capability to create and modify users and user groups is being added to the list of allowed commands.

[](javascript:PopImage('IMG_307','http://images.books24x7.com/bookimages/id_30903/12fig22_alt.jpg','534','362'))  
Add a note hereFigure 12-22: Selecting which types of commands can be run by members of this user group.

Add a note hereFor each option, you can select it in the drop-down and click Info to get precise information about which options are included in a given category. By combining the appropriate set of commands for a given user group, you can create a comprehensive permission profile between both the WebGUI and CLI. For example, an account manager will likely need permission to the Users and Groups and RBM command groups.

**Add a note here****Domain-Based CLI User Access**

Add a note hereOnce general categories of commands have been applied to a given user group, specific users within that group can be given access to a set of application domains in which to perform those operations. Navigate to Administration→Configuration→Application Domain to modify these settings. Choose a domain and click the CLI Access tab to begin the process of adding users.

Add a note hereSelect local users from the drop-down list, and click Add to allow those users to run CLI operations in this domain. Figure 12-23 shows a group of users that have been given access. These users will *only* be able to run the commands granted to them by the command groups in the previous section.

[](javascript:PopImage('IMG_308','http://images.books24x7.com/bookimages/id_30903/12fig23.jpg','449','83'))  
Add a note hereFigure 12-23: Defining the users who can access this domain from the command line (CLI).

## Summary

Add a note hereThe device administrator’s job is now largely complete from a management perspective, after defining the domain structure on the device and creating users with appropriate permissions to act on services within those domains. From this point on, additional administrative tasks such as network configuration, setup of logging/monitoring, backup and deployment of environments, and service development can all be delegated to their responsible parties. The permissions granted to those users, whether they are from the networking, security, production management, or development teams, will allow them to log in to the device, perform operations specific to their bailiwick, and not interfere with any other users, all while staying within the boundaries of predefined corporate governance policies