NetApp® SANtricity® Web Services Proxy 2.12

User Guide

April 2018 | 215-11802_D0
doccomments@netapp.com

NetApp®

Table of Contents

Overview of the NetApp Web Services Proxy

The NetApp Web Services Proxy (i.e., WSP) provides access through standard HTTPS mechanisms for configuring management services for E-Series NetApp storage arrays. You can install the Web Services Proxy on both Linux machines and Windows machines. Because the NetApp Web Services Proxy satisfies client requests by collecting data or executing configuration change requests to a target storage array, the NetApp Web Services Proxy module issues SYMbol requests to the target storage arrays.

The NetApp Web Services Proxy provides a Representative State Transfer (REST)-style application programming interface (API) for managing NetApp storage array controllers. The API enables you to integrate array management into other applications or ecosystems.

New in this Release

This release of the Web Services Proxy features the following enhancements:

- Updated support for SANtricity OS 11.40.2/8.40.20
- Added standard deviation of response times for all analyzed statistics

Abbreviations, Acronyms, Terms, and Definitions

The following table shows the abbreviations, acronyms, and terms used in this guide and their definitions.

Abbreviations, Acronyms, Terms	Definitions
API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
FDR	Fourteen Data Rate
JSON	JavaScript Object Notation
REST	Representational State Transfer

NetApp Web Services Proxy Interfaces

The Web Services Proxy provides REST-style interface for accessing common configuration operations and to retrieving basic configuration data, status, and statistics. For more information about the interface, go to the NetApp Web Services Proxy Developer Guide at `https://<nnn.nnn.nnn.nnn>:8443/docs`, where `nnn.nnn.nnn.nnn` represents the host server.

NetApp Web Services Proxy APIs

The Storage Management Web Services Proxy runs commands on the target controller. The REST-style API enables you to manage storage system objects, including:

- MEL events
- Disk drives
- Storage pools
- Volume copy jobs
- Snapshot groups
- Host groups
- LUN Mapping
- Volume I/O statistics
- Snapshot images
- Host groups
- Thin-provisioned volumes
- Volume mappings
- Hardware inventory
- Volume Statistics
- Snapshot volumes
- Host types
- Volumes
- Hosts
- Storage arrays
- Disk statistics

For a complete list of all endpoints, see the API documentation. You can access the API documentation at http://localhost:8080/docs/rest/index.html. The API documentation is fully interactive, allowing you to view details and perform various operations for the available endpoints. Detailed information for most endpoints is accessible through the Model section. The Model section contains information about possible values, types, and whether the field is optional. You can access an overview of endpoint data through the Model Schema section. Most endpoints under the API documentation support the GET, POST, and DELETE verbs. In addition, with proper authentication you can exercise the complete API from the documentation.

Cross-Domain Resource Sharing

Cross-Domain Resource Sharing (CORS) is handled by a `cors.cfg` file in the `working` directory of the web server as specified in the `wsconfig.xml` file. Because the CORS configuration is open by default, cross-domain access is not restricted.

To restrict Cross-Origin Resource Sharing (CORS) access, you can install and configure the optional `cors.cfg` file. For more information about the `cors.cfg` file, go to Configuring the Optional cors.cfg File.

NOTE: If no configuration file is present, CORS is open.

Symbol Web

Symbol Web is a URL in the REST API, but it gives access to almost all symbol calls. The symbol function is the part of the following URL:

`http://host:port/devmgr/storage-system/storage array ID/symbol/symbol function`

Compatible Storage Arrays and Controller Firmware

For a complete and up-to-date listing of all compatible storage arrays and firmware for the SANtricity plug-in, refer to the NetApp Interoperability Matrix Tool.

IP Support

Web Services Proxy supports both the IPv4 protocol and the IPv6 protocol.

NOTE: The IPv6 protocol might not work in some situations when the Web Services Proxy is attempting to automatically discover management address from the controller configuration, such as in IP address forwarding or when IPv6 is enabled on the storage arrays but not on the server.

NVSRAM File Name Constraints

The Web Services Proxy uses NVSRAM file names to identify version information accurately. Therefore, you cannot change NVSRAM filenames when they are to be used with the Web Services Proxy. The Web Services Proxy might not recognize a renamed NVSRAM file as a valid firmware file.

Web Services Proxy Configuration Files

After you have installed the NetApp Web Service, you can either accept the default NetApp Web Services Proxy settings or modify them to meet the unique operating and performance requirements for your environment.

Default Configuration Files

The Web Services Proxy installs the following two default configuration files:

- `wsconfig.xml`
- `users.properties`

By default, the files are installed in the following locations:

- Windows – `C:\Program Files\NetApp\SANtricity Web Services Proxy`
- Linux – `/opt/netapp/ santricity_web_services_proxy`

The following table shows the default locations and configuration files.

Default Directory Locations	Description
`<install root>/wsconfig.xml`	The primary configuration file for the Web Services Proxy
`<install root>/data/config/users.properties`	Web Services Proxy password files. For more information, refer to the following: • Configuring the users.properties File • Configuring the Optional cors.cfg File • Configuring ASUP Delivery Type • Enabling and Disabling ASUP Post-Web Services Installation

Configuring the optional cors.cfg File

Cross-Domain Resource Sharing (CORS) is handled by the `cors.cfg` file in the working directory in the web service, as specified by the `wsconfig.xml` file. The CORS configuration is open by default, so cross-domain access is not restricted. If no configuration file is present, CORS is open. If the `cors.cfg` file is present, it is used. If the `cors.cfg` file is empty, you cannot make a CORS request.

To configure CORS settings, add lines to the `cors.cfg` file. Each line in the CORS configuration file is a regular expression pattern to match. The origin header must match a line in the `cors.cfg` file. If any line pattern matches the origin header, the request is allowed. The complete origin is compared, not just the host element. This allows requests to be matched not only on the host, but also according to protocol, such as the following:

- Match localhost with any protocol—*localhost*
- Match localhost for HTTPS only—https://localhost*

Configuring the wsconfig.xml File

The `wsconfig.xml` file controls most of the service. Use the `wsconfig.xml` to configure the HTTP and HTTPS ports and various directory paths.

Configuring Polling Intervals

The Web Services proxy provides access to both raw storage array statistics as well as analyzed array statistics. The raw statistics provide the total counters for the various data points at the time of data collection. Raw statistics can be used for things like total read operations or total write operations. The analyzed statistics provided calculated information for an interval. Examples of analyzed statistics are read input/output operations (IOPs) per second or write throughput.

To gather statistics from storage arrays configured on the proxy, you must specify a polling interval in seconds.

To enable statistics polling, add the following line to the `wsconfig.xml` file inside the `<env-entries>` and `</env-entries>` tags, where n is the number of seconds for the interval between polling requests:

```
<env key="stats.poll.interval">n</env>
```

Example

```
<env-entries>
<env key="stats.poll.interval">60</env>
</env-entries>
```

- Polling starts at 60-second intervals; that is, the system requests that polling starts 60 seconds after the prior polling period was completed, regardless of the duration of the prior polling period. It does *not* mean that polling starts every 60 seconds.
- All the statistics are time-stamped with the exact time they were retrieved. The system uses the time stamp or time difference on which to base the 60-second calculation.

NOTE: Because the statistics are cached in memory, you might see an increase of about 1.5 MB of memory-use for each array.

Resolving Port Conflicts

When the Web Services Proxy is running, but another application is available at a defined address or port, a port conflict can occur.

1. Change the port or ports configured in the `wsconfig.xml` file.

Example

```
<sslport clientauth=”request”>8443</sslport>
<port>8080</port>
```

2. Restart the service.

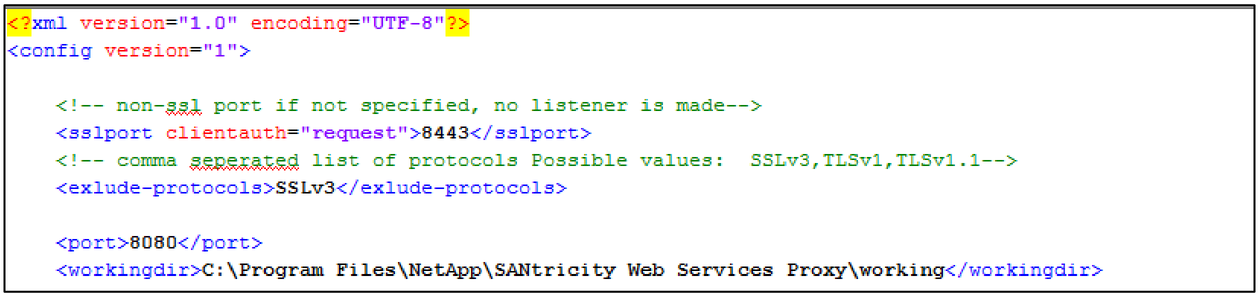
The following table shows the attributes of the NetApp Web Server configuration file that control HTTP ports and HTTPS ports.

Name	Description	Parent Node	Attributes	Required
`config`	The root node for the config	Null	Version - The version of the config schema is currently 1.0.	Yes
`sslport`	The TCP port to listen for SSL requests. Defaults to 8443.	config	Clientauth	No
`port`	The TCP port to listen for HTTP request, defaults to 8080.	config	-	No

To configure the `wsconfig.xml` file, perform these actions:

1. Within a text editor, configure the `<install root>/wsconfig.xml` file.
2. Make the necessary changes.

3. Save the file.
4. Close the file.
5. Restart the service.
   The following screenshot displays an example of a sample screen output of the `wsconfig.xml` file.



```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1">

    <!-- non-ssl port if not specified, no listener is made-->
    <sslport clientauth="request">8443</sslport>
    <!-- comma seperated list of protocols Possible values:  SSLv3,TLSv1,TLSv1.1-->
    <exlude-protocols>SSLv3</exlude-protocols>

    <port>8080</port>
    <workingdir>C:\Program Files\NetApp\SANtricity Web Services Proxy\working</workingdir>
```

Configuring the users.properties file

The `users.properties` file contains user authentication information, including user names, passwords, and roles. The file is in the `<install root>/data/config` directory by default. You can modify the directory's default install root as needed. For detailed information about user names, passwords, and roles, go to User Roles and Access.

When you edit the `users.properties` file, type the password as plain text. Then use the `securepasswds` command line utility to encrypt the passwords. The utility is installed in the base install directory for the Web Services Proxy.

With the Web Services Proxy 2.0 release, SHA256 encryption is applied to passwords under the `users.properties` file. Prior to the Web Services Proxy 2.0 release, passwords were encrypted through MD5 hashing. These MD5 encrypted passwords retain this encoding and are still valid under the `users.properties` file. However, MD5 encrypted passwords are not as secure as those passwords with SHA256 encryption.

If needed, you can apply SHA256 encryption to existing MD5 encrypted passwords under the `users.properties` file. To apply SHA256 encryption to an existing password, re-enter the MD5 encrypted password as plain text under the `users.properties` file and then run the `securepasswds` command line utility to re-encrypt the password.

The following is an example of contents under the `users.properties` file with SHA256 encoding.

```
admin=SHA256\:8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918,
security.admin,storage.admin,storage.monitor,support.admin

ro=SHA256\:7ef9ec0cf2c4facafddd03ab96eca0939d6749b49952bd816f1e0cc6901941d5,sto
rage.monitor

rw=SHA256\:55e1ebd3ebe4f1b46a5ccc9866df4d74e99fe240397e155d04664e5ce2d8e5dc,sto
rage.admin,storage.monitor,support.admin
```

User Roles and Access

User access to the NetApp Web Services Proxy is based on user roles and their corresponding levels. Only the Read-Write user role can access the Array Manager and the array tree. The Read-Write role enables you to perform any action to a storage array in the array tree in the Array Manager.

- The initial username and role is `rw.`
- The password is `rw.`

The following file contains the user IDs, user roles, and passwords:

```
<install root>/data/config/users.properties
```

User names, passwords, and roles are in the following sequence:

```
user=encryptedpassword,storage.role
```

Role-based access control

With the 2.1 release of the NetApp Web Services Proxy, new roles are available to provide more granular access control. All pre-2.1 WSP roles are now deprecated. However, most pre-2.1 WSP roles should continue to function as previously configured with the exception of the `storage.admin` role. The roles introduced in 2.1 are not additive in their permissions and multiple roles must be specified for a single user to allow for the same level of access. Role requirements for each endpoint is available through the API documentation.

Refer to the following tables for which roles are now required in the configuration files.

Available roles for Web Services Proxy 2.1

Role	General Description
`security.admin`	SSL Management
`storage.admin`	Storage array configuration
`storage.monitor`	Viewing storage array related data
`support.admin`	Special role for support operations such as ASUP retrieval.

Pre-2.1 WSP to 2.1 WSP Conversion table

Pre-2.1 WSP Role	2.1 WSP Roles Required
`storage.admin` (or `admin`)	`security.admin`
	`storage.admin`
	`storage.monitor`
	`support.admin`
`storage.rw`	`storage.admin`
	`storage.monitor`
	`support.admin`
`storage.ro`	`storage.monitor`

Flags and Settings

You can edit the following other settings in the Environment Entries section.

```
<env-entries>
```

```
   <!-- Enables basic authentication. The user no longer is required to use the
   /devmgr/utils/login URL --->
```

```
<env key="enable-basic-auth">true</env>
```

Enabling LDAP functionality

LDAP integration allows for user authentication and mapping of roles to LDAP groups. A dedicated `ldap.xml` configuration file is required to manage the user attributes and group mappings used for LDAP. Enabling the LDAP function and the loading of the `ldap.xml` file requires the following `<userauth>` entry as a child element of `<config>` under the `wsconfig.xml` file:

```
<userauth>ldap</userauth>
```

NOTE: Remove any existing values under `<userauth>` before configuring the element for LDAP use.

For more information on the `ldap.xml` file, refer to Managing the ldap.xml configuration file.

Logging in to Web Services Proxy

Login URL Authentication

This is the default way to log in. The sample code shows, using the cookie that it is set on, when the `/util/login` URL is used.

For reference, the cookie value to pass back to the server is `JSESSIONID`.

Basic Authentication

You can use basic authentication when it is enabled. If you are not logged in, the server returns a basic authentication challenge. To enable basic authentication, add the following lines to the `wsconfig.xml` file.

```
<env-entries>
<env key="enable-basic-auth">true</env>
</env-entries>
```

NetApp Web Services Proxy Security

The NetApp Web Services Proxy uses Secure Sockets Layer (SSL) for security.

Generating a Self-Signed Certificate

To enable SSL, add an SSL port designation to the `wsconfig.xml` configuration file. When the server is started with SSL configured, the server looks for the keystore and truststore files.

- If the server does not find a keystore, the server uses the IP address of the first non-loop back IPv4 address that it finds to generate a keystore and add a self-signed certificate to the keystore.
- If the server does not find a truststore, or the truststore is not specified, the server uses the keystore as the truststore.

Generating an SSL Certificate

The NetApp Web Services Proxy provides a Java keytool with which to generate an SSL certificate. You can generate a signed SSL certificate and export and store it on each client.

Generating an SSL Certificate on the Application Server

After you have generated the certificate and saved it in the application server keystore, you can use the certificate again on the same application server.

1. Remove any auto-generated keystores in the working directory.
2. Stop the server.
3. Run the following command to generate the certificate:

```
keytool -genkeypair -keyalg RSA -keysize 2048 -alias jetty -dname CN=<THE SERVER
DNS NAME> -keypass changeit -storepass changeit -keystore keystore -ext
san=ip:<THEIR IP ADDRESS>,dns:<THE SERVER DNS NAME> -validity 999
<or>
keytool -genkeypair -keyalg RSA -keysize 2048 -alias jetty -dname CN=servername -
keypass changeit -storepass changeit -keystore keystore -ext
san=ip:192.168.1.1,dns:servername -validity 999
```

   The following message appears in the terminal window:

```
When prompted for a password, use "changeit", unless you specify a specific one in
the wsconfig.xml file
When prompted for your first and last name, use the IP address or DNS name of the
host, whichever one you plan on using in URLs
```

4. Follow the instructions in the terminal window.
5. Run the following command to export the certificate for signing:

```
keytool -certreq -alias jetty -file mycertreq.cet -keystore keystore -dname
CN=servername -ext san=ip:192.168.1.1,dns:servername
```

6. Send the certificate request to a certifying authority to be signed.
7. Run the following commands to import the CA certificate and the signed certificate back into your keystore.

```
keytool -import -trustcacerts -alias root -file <CA CERT FILE> -keystore keystore
keytool -import -trustcacerts -alias jetty -file <signed cert from ca> -keystore
keystore
```

8. Restart the server.
9. Save the certificate in your keystore.

Generating an SSL Certificate on an Application Client

If you do not already have the certificate, import it from the certifying authority. Follow the prescribed import process for your specific operating system and web browser.

Stateless mode

The Web Services proxy provides a stateless mode option of the REST API. The stateless mode allows arrays to access the Web Services proxy without prior registration. Information normally supplied during registration is supplied with each call through the stateless mode. Stateless arrays are cached in a stateless cache map within the proxy to provide a multi-call performance approaching that of registered arrays.

Due to state and database dependency requirements, the following functionality is not available for stateless arrays:

- Display in array lists
- The ability to be added to folders
- Analyzed, cached, or historical stats
- Cached MEL events
- Proxy-based volume or system tagging

Stateless Access Token

Each call made through the stateless mode requires a unique security token. Self-generated tokens can be used for these calls or you can obtain a token through the REST endpoint `GET_/v2/client-token` under `Administration`. The token is required for use as the ID for the array in the stateless cache map and as the key in the map of client IDs to a map of compound keys to storage devices and. For general information on how to access REST endpoints, refer to NetApp Web Services Proxy APIs.

General workflow

Calls for stateless mode are located under the `v2/storage-systems` URL. A storage system ID of `stateless` is required when using stateless mode. Whenever the `ArrayDataHandler` identifies the `stateless` storage system ID, a query of the request object for the following other headers is performed:

- `x-netapp.mgr-paths` – A required header containing a comma separated list of IP or DNS names for the controllers and/or agent to access the storage systems.
- `x-netapp-webapi-client-token` – A required header containing a token used exclusively to identify a client. It is recommended you use a cryptographically random token. The token in this header keeps arrays isolated.
- `x-netapp-sa-password` – An optional header for `GET` operations, the string value for the SA password. Although optional, `GET` operations will fail if a storage array password is set and this header is not set to the correct password.
- `x-netapp-wwn` – An optional header containing the array WWN. A device ID/WWN is not required for out-of-band.
- `x-netapp-system-create-timeout` – An optional header containing the number of seconds to wait for a storage system to come online before returning offline.

After the headers are queried, the stateless cache map for stateless arrays is examined. The data structures is a map of client ID to a map of Storage ID (Management Paths + WWN) to Storage System. If the system is not found, a new system is created. The password is set on the storage system for each call, and the stateless storage system is returned to handle all calls. After being added to the cache map, the client cache entry or storage device cache entry are removed if either are not accessed during the timeout period.

Configuring stateless.cache.expire setting

If needed, you can configure the expiration setting for the cache of the stateless mode storage system through `stateless.cache.expire` setting under the `wsconfig.xml` file. By default, the stateless cache expires in 300 seconds if not accessed.

Adding Storage Arrays

Automatically Discovering Storage Arrays

By default, you need to provide only one management IP or DNS address to add an array. The server automatically discovers all management paths when the paths are not configured or they are configured and rotatable.

NOTE: If you attempt to use an IPv6 protocol to automatically discover storage arrays from the controller configuration after an initial connection has been made, the process might fail. Possible causes for the failure include problems during IP address forwarding or IPv6 being enabled on the Storage Systems but not being enabled on the server.

Turning Off Automatic Discovery of Storage Arrays

When the paths are configured, but not configured so that the server can route to the addresses, intermittent connection errors happen. If you cannot set the IP addresses to be routable from the host, you can turn off auto discovery. To turn off auto discovery, modify the following lines in the `wsconfig.xml` file.

```
<env key="autodiscover.ipv6.enable">false</env>
<env key="autodiscover.ipv4.enable">false</env>
```

Automatic Polling of Volume and Disk Statistics

You can use the REST service to set up an automatic polling and caching of volume and disk statistics. To enable automatic polling, modify the `wsconfig.xml` file normally located in the `webserver` directory. The new service polls for all disk and volume statistics on the storage array registered with the service.

There are two different types of statistics APIs that we provide raw and analyzed statistics. Raw statistics are linear in nature and typically require at least two different collected data-points to derive usable data from them. The analyzed statistics are a derivation of the raw statistics that provide the metrics we believe are most important to our users. Many of the values that can be derived from the raw statistics are present in a usable, point-in-time format in the analyzed statistics for your convenience.

The raw statistics may be retrieved regardless of whether the automatic polling is enabled for statistics. If enabled, you may however add the `usecache=true` query string to the end of the URL to retrieve cached statistics from the last poll. Using cached results greatly increases the performance of statistics retrieval. However, multiple calls at a rate equal to or less than the configured polling interval cache will retrieve the same data.

2.1 Statistics Functionality

With the 2.1 release of the SANtricity Web Services Proxy, four new APIs were introduced that allow for the retrieval of raw and analyzed controller and interface statistics from supported hardware models and software versions. These statistic APIs are available for any 28xx or newer systems, as well as for any model 27xx or 56xx systems that are running software versions 08.30.20.xx/11.30.20.xx or newer. Among other metrics, the controller statistics APIs provide CPU statistics.

Raw Statistics APIs

- /storage-systems/{system-id}/controller-statistics
- /storage-systems/{system-id}/drive-statistics/{optional list of disk ids}
- /storage-systems/{system-id}/interface-statistics/{optional list of interface ids}
- /storage-systems/{system-id}/volume-statistics/{optional list of volume ids}

Analyzed Statistics APIs

- /storage-systems/{id}/analysed-controller-statistics/
- /storage-systems/{id}/analysed-drive-statistics/{optional list of disk ids}
- /storage-systems/{id}/analysed-interface-statistics/{optional list of interface ids}
- /storage-systems/{id}/analysed-volume-statistics/{optional list of volume ids}

These URLs retrieve analyzed statistics from the last poll and are only available when polling is enabled. These URLs include the following input-output data:

- Operations per second
- Throughput in megabytes per second
- Response times in milliseconds

These calculations are based on the differences between statistical polling iterations, which are the most common measures of storage performance. These statistics are preferable to unanalyzed statistics.

NOTE: When the system starts, there is no previous statistics collection to use to calculate the various metrics, so analyzed statistics will require at least one polling cycle after startup to return data. In addition, if the cumulative counters are reset, the next polling cycle will have unpredictable numbers for the data.

AutoSupport (ASUP)

The AutoSupport (ASUP) feature collects data in a customer support bundle and automatically sends the message file to technical support for remote troubleshooting and problem analysis. ASUP automatically transmits messages to NetApp based on manual and schedule based criteria. Each ASUP message is a collection of log files, configuration data, state data, and performance metrics.

The ASUP feature transmits the following files to the NetApp technical support team:

File Name	Description
`x-headers-data.txt`	A .txt file containing the X-header information
`manifest.xml`	An .xml file detailing the contents of the message
`arraydata.xml`	An .xml file containing the list of client persisted data
`appserver-config.txt`	A .txt file containing the web server configuration data
`wsconfig.txt`	A .txt file containing the web server configuration data.
`host-info.txt`	A .txt file containing information about the host environment

File Name	Description
`server-logs.7z`	A .7z file containing every available webserver log file
`client-info.txt`	A .txt file with arbitrary key/value pairs for application-specific counters such as method and webpage hits

ASUP Schedule

By default, ASUP transmits data once a week.

Configuring ASUP Delivery Type

HTTPS is the default delivery method for the ASUP feature. Users can configure the ASUP feature to use HTTPS, HTTP, or SMTP delivery methods through the `ASUPConfig.xml` file. To modify the ASUP delivery method, enter one of the following values under `<delivery type=”(integer)”>` of the `ASUPConfig.xml` file:

- 1 – The default delivery method for the ASUP feature, delivers ASUP data via HTTPS

  ```
  <delivery type=”1”>
  ```

- 2 – Delivers ASUP data via HTTP

  ```
  <delivery type=”2”>
  ```

- 3 – Delivers ASUP data via SMTP. To properly configure the ASUP delivery type to SMTP, enter the following command under the `ASUPConfig.xml` file:

  ```
  <delivery type=”3”>

  <smtp>

  <mailserver>smtp.example.com</mailserver>

  <sender>user@example.com</sender>

  <replyto>user@example.com</replyto>

  </smtp>

  </delivery>
  ```

Enabling and Disabling ASUP Post-Web Services Installation

The ASUP feature can be enabled or disabled during the initial installation of the Web Services Proxy. If needed, users can enable or disable the ASUP feature post-Web Services installation through the `ASUPConfig.xml` file. To enable or disable the ASUP feature post Web Services installation, enter one of the values under `<asupdata enabled=”(Boolean)” timestamp=”1428601077263”>` of the `ASUPConfig.xml` file:

- `true` – Enables the ASUP feature

  ```
  <asupdata enabled=”true” timestamp=”0”>
  ```

- `false` – Disables the ASUP feature

  `<asupdata enabled=”false” timestamp=”0”>`

NOTE: The `timestamp` entry is superfluous.

For more information on enabling and disabling the ASUP feature during the Web Services Proxy installation process, refer to the *NetApp SANtricity Web Services Proxy Install Guide*.

Managing Auto Updates

Starting with version 1.2, the product can automatically download updates that can be installed the next time the application restarts or on demand using the REST API. You can enable or disable this feature after installation by editing the `wsconfig.xml` file. The following key controls the auto update feature: `<enable-auto-update>true</enable-auto-update>`

The value can be `true` or `false`.

When the above value is set to `true`, the software checks for updates, and if there is an update, downloads it. The software checks once a day. The overhead for the checking is extremely low.

To manually download updates and install updates at runtime, use the REST API. For further details, see the REST documentation for the URL v2/upgrade.

Updates are downloaded in the background. The download should have no negative effect on the system.

After downloading, installing the updates at runtime takes several seconds, causing an interruption in service. This does not happen automatically and only occurs when you request it or on restart.

Rolling back to original software

The updated software is placed in the directory `<install root>/working/webapps`.

If you want to go back to the originally shipped software, you can remove the war files (*.war) from that directory and either restart the application or use the REST API to reload the software.

The original software is always saved in `<install root>/data/webapps`. Do not remove or replace files in this directory.

Version Numbers

There are two version numbers associated with the product:

- The version of the application (01.20.XXXX.XXX)
- The version of the REST API

Version number reported via the /utils/about URL and reported in the upgrade URL is the version for the REST API. Unless you install a new version of the application with the installer, the application version number does not change. An auto update will upgrade the REST API component.

Logging in to the API

Web Services Proxy has two default user logins and permission levels:

- Read-write access
  - User ID: rw
  - Password: rw
- Read-only access
  - User ID: ro
  - Password: ro

To log in, type the following URL in a web browser:

```
http://<host:port>/utils/login
In addition, the user can use “Basic Authentication” to login to the service. If a
login session has not been established. A Basic Authentication challenge will be sent
to the client.
```

Scaling Up the Number of Managed Arrays

The default setting for the Web API can handle up to 100 storage systems. If you need to manage more, you must bump the memory requirements for the server. On Windows, this is handled in the appserver64.init file.

Change the line `vmarg.3=-Xmx512M`.

On Linux, the line is in the webserver.sh look for the line `JAVA_OPTIONS="-Xmx512M"`.

To increase the memory, add 250 MB per 100 extra storage arrays. Do not add more memory than what you physically have and allow enough extra for your operating system and other applications.

In addition to memory, the application uses network ports for each storage system. Linux and Windows consider network ports as file handles. Because of this, see the section on increasing file handles to allow for more storage systems.

MEL Events Cache Size

The default cache size is 8192 events. The approximate data usage for the MEL events cache is 1MB for each 8192 events. Therefore, by retaining the defaults, cache usage should be approximately 1MB for a storage array.

File Handles Limit

As a security measure, most operating systems limit the number of open file handles that a process or a user can have open at one time. Especially in Linux environments, where open TCP connections are considered to be file handles, it is very easy for the Web Services Proxy to exceed this limit. Because the fix is system dependent, you should refer to your operating system’s documentation for how to raise this value.

Configuration for Load-balancing and/or High-Availability

If needed, you can use the Web Services Proxy in a highly-available (HA) configuration. In an HA configuration, typically either a single node will receive all requests while the others are on stand-by, or requests will be load-balanced across all nodes.

The Web Services Proxy is capable of existing in a highly-available (HA) environment, with most API's operating correctly regardless of the recipient of the request. Metadata tags and folders are two exceptions to that rule, as they are currently stored in a local database and are not shared between Web Services Proxy instances.

However, there are some known timing issues that exist and will manifest in a small percentage of requests. Specifically, it is possible for one instance of the proxy to have newer data faster than a second instance, for a small window. We have added a special configuration option to the Web Services Proxy that has been tested and shown to remove this issue.

This option is not enabled by default, as it will cause an increase in the amount of time it takes to service requests (for data consistency).

To enable this option, it is currently necessary to add a system property to the `webserver.sh` file.

```
DEBUG_START_OPTIONS="-Dload-balance.enabled=true"
```

Managing the ldap.xml configuration file

Once loaded through the wsconfig.xml file, the ldap.xml configuration file resides in the working directory. The ldap.xml configuration file stores the various user attributes and group mappings that are used to query LDAP. The `ldap.xml` file configures how the REST API queries LDAP for authenticating user identification and determining group membership for role mapping.

NOTE: You must restart the server for any changes made to the ldap.xml file to take effect.

Copy the `ldap.xml` file from `<install directory>/samples/ldap` to the working directory (typically `<install directory>/working`). The file includes documentation and examples on how to configure the file appropriately. Subsequent topics of this section detail how to modify each segment of the file.

Example of ldap.xml configuration file

```
<ldapconfig version="1">
    <ldap-domain>
        <name>ldap.example.com</name>
        <name>ldap</name>

        <url>ldaps://ldap.hq.example.com:636</url>

        <login-base>
            cn=%s,ou=users,dc=example,dc=com
        </login-base>

        <user-role-mechanism>mixed</user-role-mechanism>

        <bind-lookup-user password="2jNID6pxSj0YgjFs">
            cn=BindUser,ou=system-accounts,dc=example,dc=com
        </bind-lookup-user>

        <search-base>ou=users,dc=example,dc=com</search-base>

        <filter-base>cn=%s</filter-base>

        <group-attributes>
            <group>memberOf</group>
            <group>managedObjects</group>
        </group-attributes>

        <group-map>
            <role name="security.admin">cn=storage-
            admin,ou=groups,dc=example,dc=com</role>
            <role name="storage.admin">cn=storage-
            owners,ou=groups,dc=example,dc=com</role>
            <role name="storage.admin">.*cn=data-admin.*</role>
            <role name="storage.monitor">.*cn=storage-viewer.*</role>
        </group-map>

        <!-- <user-role-file>user-roles.properties</user-role-file> -->

        <default-group-role>storage.monitor</default-group-role>
        <default-group-role>storage.admin</default-group-role>
    </ldap-domain>
</ldapconfig>
```

Security Considerations

When configuring security for the LDAP service, consider the following items:

- For the highest security, avoid using wildcard pattern `.*` in group-map role definitions of LDAP group DNs.
  - If wildcard patterns are used, be certain not to use patterns that could inadvertently match group names that should not be mapped to roles.
- Local users can be disabled by removing their data from the `data/config/user.properties` file.

Domains

You can specify multiple domains in a single ldap.xml configuration file. Each domain must be listed with an ldap-domain entry.

You can name domains anything as long as they are valid DNS names containing only the ASCII letters ‘a’ through ‘z’ (in a case-insensitive manner), the digits ‘`0`’ through ‘9’ and the hyphen (‘`-`‘). However, the digits under the DNS name cannot start with a hyphen. An LDAP domain can have multiple names to allow multiple possible names to be specified when logging in.

Example

```
<ldap-domain>
     <name>ldap.example.com</name>
     <name>example</name>
```

NOTE: DNS-resolvable names are not required for domain names under the `ldap-domain` entry.

After you enable LDAP, Web Services requires the username used for logging in be specified as follows:

```
<username>@<ldap-domain>
```

The specified `<username>` is used under the `filter-base`, `login-base`, and `user-role-file` elements of the ldap.xml file for determining the role mapping of the user. The specified `<ldap-domain>` is simply the name of the defined domain.

Local Domain

There is a special, reserved local domain. When a user/login name is specified as `<user>@local`, it references the local domain which are the contents of the `users.properties` file. Authentication and role assignment take place using this legacy mechanism.

For example, a user may login as `rw@local`. If the `users.properties` file contains an `rw` user and the supplied password, the user is authenticated and given the role specified for that user in the `users.properties` file.

NOTE: For a summary of REST API roles and access, refer to User Roles and Access and Logging in to the API.

LDAP URL

The URL for the LDAP service is configurable under the `URL` element of the ldap.xml file. The LDAP URL entry must be specified as either `ldap` or `ldaps` protocol and contain the IP address. In addition, the port for the LDAP URL must be specified (typically 389 for `ldap` and 636 for `ldaps`).

Example

```
<url>ldaps://ldap.example.com:636</url>
```

NOTE: If using ldaps (recommended), be sure to use a host name or IP address for which the server certificate was signed.

Adding a certificate to the trust store

If using `ldaps`, you must add a certificate to the trust store for the LDAP service. Before you restart the server to enable the ldap.xml file changes, you must complete the following steps.

1. Obtain one of the following as a Base-64 encoded X.509 file:
   - The root CA certificate that signed the LDAP server certificate (preferred)

     -Or-
   - The LDAP server’s self-signed certificate
2. From a command prompt, navigate to the install directory.

   Example (Linux)

   ```
   cd /opt/netapp/santricity_web_services_proxy
   ```

   Example (Windows)

   ```
   cd "C:\Program Files\netapp\SANtricity Web Services Proxy"
   ```
3. Add the certificate to the webserver truststore keystore file in `working/truststore`.

a. Run the following command:

```
jre/bin/keytool -importcert -file <certificate path> -keystore
working/truststore
```

   i. The `<certificate path>` is the path to the CA server or server certificate file.

   ii. Unless changed, the default password is `changeit`.

b. Enter `yes` to confirm the import.

Login Base customization

Optionally, the login pattern used with the user login name is configurable through the login-base entry. If the login-base is defined, the login-base value is used as a pattern for the directory user DN for binding to the LDAP server for validating the username and password. The `%s` is substituted with the user-specified username after escaping special DN characters. If the login-base is not defined, the user-supplied username itself is used as the DN for binding to the LDAP server for validating the username and password.

Example

```
<login-base>cn=%s,ou=users,dc=example,dc=com</login-base>
```

Modes of Granting User Roles

You can configure one of four modes for granting user roles configured under the ldap.xml file. Refer to the following for a brief overview of each available mode of granting user role:

- `ldap-only` – Looks for group membership in LDAP for role mapping. If nothing is found, yet the user is authenticated, the user receives the default role if defined. If a group membership, role mapping, and default role is not defined, a role is not provided and the login will fail. Filter base, search base, and group-mappings must be defined in the ldap.xml file for this user role.

- `file-only` – After the user is authenticated, the user’s ID is searched for in the role properties file and the specified role is assigned. If the role properties are not located in the role properties file, the user receives the default role if defined. The role file name is definable through the `user-role-file` entry under the ldap.xml file. For information on how to configure the user-role file, refer to User Role File.

- `mixed` – Looks for user role in LDAP and grants this role and then looks for the user role specified in the role file. Any user role specified under the user-role-file entry overrides any user role assigned through the LDAP lookup. The `mixed` user role is essentially a combination of `ldap-only` and `file-only`. Filter base, search base, and group-mappings must be defined in the ldap.xml file for this user role. The role file name is definable through the `user-role-file` entry under the ldap.xml file. For information on how to configure the user-role file, refer to User Role File.

- `default-role` – After the user is authenticated through LDAP, the default role configured through the ldap.xml file is used. The `default-group-role` entry must be configured under the ldap.xml file for this user role.

The mode of granted user role is configurable through the `user-role-mechanism` entry in the ldap.xml file.

Example

```
<user-role-mechanism>mixed</user-role-mechanism>
```

NOTE: Regardless of the specified `user-role-mechanism` value, the user's password is validated through LDAP bind operation.

Bind Lookup User

If needed, you can configure a specific user to use when looking up the group membership for users. Typically, you would configure the `bind-lookup-user` entry within the ldap.xml file whenever regular users might lack reader permissions to view their own group membership. The bind user information must be specified as a full DN.

Example

```
<bind-lookup-user password="2jNID6pxSj0YgjFs">
       cn=BindUser,ou=system-accounts,dc=example,dc=com
</bind-lookup-user>
```

Search Base and Filter Base customization

The search-base within the ldap.xml file is used to find group memberships of the user. The search-base is the DN in the directory of a container object of users. The filter-base is used to find the user object within this container. After the user object is located, any associated group membership is identified.

Example

```
<search-base>ou=users,dc=example,dc=com</search-base>
```

A template string for the LDAP search filter is definable through the `filter-base`. The `filter-base` within the ldap.xml file allows for variable substitution of the user-supplied username in the search string. To enable this search, a single variable string defined by `%s` in place of the special character escaped user-supplied username must be configured through the filter-base entry under the ldap.xml file.

Example

```
<filter-base>userPrincipalName=%s</filter-base>
```

Group mappings

The mapping of groups to roles is configurable through the `group-map` entry under the ldap.xml file. The `role` element's `name` attribute specifies the role. The element value specifies the DN of the group. The wildcard `.*` can optionally be used to allow specifying only a partial DN in the `ldap.xml` file.

Example

```
<group-map>
     <role name="security.admin">cn=storage-admin,ou=groups,dc=example,dc=com</role>
     <role name="storage.admin">cn=storage-owners,ou=groups,dc=example,dc=com</role>
     <role name="storage.admin">.*cn=data-admin.*</role>
     <role name="storage.monitor">.*cn=storage-viewer.*</role>
</group-map>
```

Attributes defining group membership is defined through the group-attribute entry.

Example

```
<group-attributes>
    <group>memberOf</group>
    <group>managedObjects</group>
</group-attributes>
```

User Role File

Whenever the `file-only` or `mixed` modes of granting user roles are configured under the `user-role-mechanism` entry, the specified user-role file are loaded to determine role mappings for users specified within.

Example

```
<user-role-file>user-roles-win1.properties</user-role-file>
```

NOTE: If the `user-role-file` element is not specified, the file `user-roles.properties` under the working directory is loaded. Otherwise, the specified file will be loaded relative to the working directory.

User Role File Contents

The user role file should be a plain text file. One username should be listed per line (`<username>` portion of `<username>@<ldap-domain>`) with an equals sign (=) separating it from the role name. Copy the sample file from `<installation directory>/samples/ldap/user-roles.properties` to the appropriate location and configure the copied file. The file includes further documentation and examples.

Example

```
aeinstein = storage.admin,security.admin,storage.monitor,support.admin
janed = storage.monitor
isaacn = storage.admin,storage.monitor,support.admin
jsmith = none
```

Default Role

A default role might be specified for users authenticated with LDAP that have no mapped LDAP group and no defined user role. Otherwise, such users are denied access. The default role is configurable through the `default-group-role` entry under the ldap.xml file and multiple entries can be specified.

Example

```
<default-group-role>storage.monitor</default-group-role>
```

LDAP Test Utility

The LDAP Test Utility is used to test and validate an LDAP configuration without starting the entire Web Services server. The ldaptest.jar file contains an LdapTest utility class.

From the installed web services directory, perform the following to run the LDAP Test Utility:

```
jre/bin/java -jar ldaptest.jar -file working/ldap.xml -user "<user name>@<domain>" -
password <password> -truststore working/truststore -trustpassword changeit
```

Notes of interest regarding the LDAP Test Utility command

- Where `<user name>` is the user name, `<domain>` is the name of the LDAP domain as defined in the ldap.xml file.

- If you prefer to be prompted for the passwords rather than specifying the data in the command, enter `–P` instead of `–password <password>` and `–T` instead of `-trustpassword <password>`. You will then be prompted for passwords and the data will be masked as entered.

- You can exclude the `–truststore` and `–trustpassword` options if you are not using `ldaps`. If you used a different trust store password from the default, substitute `changeit` with the actual password used.

- When running the LDAP Test Utility, verify you receive the expected results and no unexpected errors are reported. It is recommended you run the utility with a variety of different user configurations to ensure thorough testing.

Parameters:

```
-file <filename> -- The LDAP .xml file name.
-user <user Id>  -- The user to login and authenticate. This can be of the form
                    <name>@<domain> or just <name>.
-password <pw>  --  The password to use for the user.
-P              --  Interactively ask for the user password which will be masked as it
                    is entered.
-truststore <filename> -- The trust store file name, may be needed for ldaps."
-trustpassword <pw>    -- The password to use for the truststore."
-T              --  Interactively ask for the truststore password which will be masked
                    as it is entered.
-usersfile      --  A users.properties file, when present will create local domain.
```

Copyright information

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

http://www.netapp.com/us/legal/netapptmlist.aspx

How to send your comments

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email. *doccomments@netapp.com*. To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277

NetApp®