Agent-XML

Contents

Agent-XML Overview

Agent XML is used when a program needs to authenticate users to Swivel. It also has functions for changing PIN and other features, but does not have the ability to change user details in the Swivel database.

Swivel can authenticate users with 2 different protocols.

- RADIUS is a standards-based protocol, for further information see RADIUS How To Guide

- AgentXML is a proprietary Swivel standard. Agent-XML allows greater control as it also allows Reporting, Admin and Helpdesk functionality.

Swivel requirements

Configuring Swivel to use Agent-XML Requests

Swivel must be configured to allow an agent to communicate, details on how to permit an agent to communicate with the Swivel server are here: Agents How to Guide

Repository and Agent configuration

AdminRequest only works with users in the repository with the same name as the corresponding Agent. The work around is to give the Repository exactly the same name as the Agent (or vice versa). From version 3.9.2 Helpdesk requests are allowed for agents that are not repositories.

If you use HelpdeskRequest instead, you can specify a repository attribute on the Read element, but AdminRequest doesn't support this.

The reason for the distinction is that the AdminXML API was designed to support creation and manipulation of users from outside Swivel. The ability to access users from other repositories was an added feature, hence the use of HelpdeskRequest, where the commands are largely read-only.

Agent-XML

The Agent-XML API is an XML based API used for integrating Swivel with other applications.

There are 4 subsets of the API, that cover the following areas

- Authentication

Authentication, Change PIN, PIN Reset, Start Authentication Session, Request security Strings etc

- Admin functions

Add new users, set user details (eg mobile phone number), synchronise a repository, delete users

- Helpdesk functions

Unlock user, send user new credentials, set user policies etc

- Reporting functions

Retrieving lists of idle users etc

All these APIs follow the structure described in this article.

Principles

The APIs are based around XML documents for both request and reply.

The basic idea is to build a document containing details of operation(s) to be performed and the user(s) on which they are to be performed, submit it to Swivel via HTTP and Swivel will reply with a document detailing which operations succeeded and which, if any, failed. All operations via the API will be logged in the standard Swivel log.

Only authorised Swivel Agents will be able to submit requests. This is in line with Agents used for authorisation. The API is case sensitive, the correct case shown in the included examples.

Requests are sent to Swivel via an HTTP POST using http or https depending on the configuration of the Swivel server. For appliances the default will be https over port 8080.

The requests are posted to the Swivel context followed by AgentXML for the authentication API and AdminXML for the Admin, Helpdesk and reporting API.

For example http://<ip address>:8080/pinsafe/AgentXML?xml=<?xml version="1.0"?>

or

https://<ip address>:8080/pinsafe/AdminXML?xml=<?xml version="1.0"?>

Structure

The structure of a request that all API requests are contained within a request element. This request element specifies the type of request and also includes the shared secret for agent authentication.

e.g. an authentication request would be contained within an SASRequest tag

```
<?xml version="1.0" ?>
<SASRequest secret="MyAdminAgent" version="3.4">
     .
     .
</SASRequest>
```

Whereas an admin API request would be contained within an AdminRequest element.

```
<?xml version="1.0" ?>
<AdminRequest secret="MyAdminAgent" version="3.4">
     .
     .
</AdminRequest>
```

The responses from the Swivel server are similarly contained within response tags eg

```
<?xml version="1.0" ?>
<SASResponse secret="MyAdminAgent" version="3.4">
     .
     .
</SASResponse>
```

and

```
<?xml version="1.0" ?>
<AdminResponse secret="MyAdminAgent" version="3.4">
     .
     .
</AdminResponse>
```

Within these elements for elements that specify the particular request. For details of these elements refer to the article relating to that part of the API.

Commands

Authentication for more information see: Authentication

The following are valid values for the <Action> element:

- changepin : Change Users Credentials
- changepassword : Change a user password (Sentry or repository)
- checkpassword : Check that a password is correct (Sentry or repository)
- exists : Check if a user have an account.
- existsbyattribute : Check if a user has an account based on custom attributes
- increaselock : Increase a user's lock count (simulate a login failure)
- getuserattribute : Return the value of a named attribute for a user
- getusernamebyattribute : Return the username given an attribute name and value
- hascachedpassword : Check if a user has an AuthControl Desktop password cached
- killsession : terminate a Sentry session
- logevent : add a message to the Sentry logs (available from 4.2.1 onwards)
- login : Perform Sentry authentication
- loginbyattribute : Perform Sentry authentication using an alternative username attribute
- oathsync : resynchronise an OATH token
- ocraverify : Verify an OATH OCRA response to challenge
- ondemandmessage : send an On Demand message
- ping : Ping Swivel application
- provision : provision a mobile application given the provision code
- provisioncode : request a provision code
- reset : Perform a PIN reset (Note: the user must have a transport to receive the new PIN)
- resetcode : Request a PIN Reset code
- sendconfirmationcode
- sessionstart : Start an authentication session
- securitystrings : Request a set of security strings
- transportindex
- validateconfirmationcode

Helpdesk for more information see: Helpdesk functions

<Reset> : Send user new credentials (Note: the user must have a transport to receive the new PIN)

<Strings> : send dual channel security string(s) to user

<Update> : Change user policies or credentials

<OathSync> : Synchronise an OATH token

<PurgeDeleted> : Purge deleted users

Admin for more information see: Admin functions

<Create> : create a new user and optionally set user details.

<Delete> : delete a user.

<Read> : read the details of a given user.

<Reset> : Send user new credentials (Note: the user must have a transport to receive the new PIN)

<Sync> : Synchronise with a repository

<Update> : Change user policies, credentials, transports etc

<PurgeDeleted> : Purge deleted users.

Reporting for more information see: Reporting functions and Reporting Using Agent-XML How to Guide

<disabled> : Report on disabled users

<idle> : Report on idle users

<locked> : Report on locked users

Agent-XML Errors

AgentXML request failed, error: The agent is not authorised to access the server.

An Agent-XML request is being made against the Swivel server but is not permitted to do so. If access should be allowed create an entry on the Swivel Administration Console under Server/Agents. If an entry exists verified the shared secret is the same on Swivel and the access device. See also Agents How to Guide. Try with the IP address instead of hostname, the hostname entry may be case sensitive

AgentXML request failed, error: The XML request sent from the agent was malformed.

The Agent XML request contains an error check the format. Spaces may also cause errors.

ADMIN_ERROR_UNSUPPORTED_VERSION

The agent version is incorrect.

https://127.0.0.1:8080/pinsafe/AdminXML?xml=<AdminRequest%20secret="secret"%20version="3.7"><Report%20repository="local"><Idle%20since="01-jul-20

Swivel 3.7 should use 3.6 for the version.

https://127.0.0.1:8080/pinsafe/AdminXML?xml=<AdminRequest%20secret="secret"%20version="3.4"><Report%20repository="local"><Idle%20since="01-jul-20

Agent XML Versions

Swivel Version	Agent-XML Version
3.7	3.6
3.6	3.6
3.5	3.4
3.4	3.4

Malformed XML

This can occur when a ../AgentXML URL request is used to send an Admin request. Use the ../AdminXML instead.

The Difference Between Admin XML and Agent XML

Admin XML is used when a program needs to make changes to the user database, for example, to add and remove users, or change their details.

Agent XML is used when a program needs to authenticate users to Swivel. It also has functions for changing PIN and other features, but does not have the ability to change user details in the Swivel database.

Both APIs can be used from the same program, and they both require that an Agent is configured on the Swivel server. You can use the same Agent for both APIs, but to use the Admin API to manage users, the Agent must have "Can act as Repository" set to "Yes".