ReportingAPI

Contents

- 1 Introduction
- 2 PRINCIPLES
    - 2.1 Sending an Admin API request
- 3 REPORTING API
    - 3.1 Disabled
    - 3.2 Idle
    - 3.3 Locked
    - 3.4 Count
    - 3.5 License information
    - 3.6 AllUsers
    - 3.7 AllUsersDetailed
- 4 Responses

Introduction

Swivel has developed an API, the User Admin API, to allow an external application to perform CRUD (Create, Read, Update, Delete) style operations on users. For most installations, these operations are performed by the User Synchronization job but for larger user populations synchronization may be impractical. The User Admin API addresses this need. In conjunction with the User Admin API, Swivel has also developed another API allowing external applications to perform the functions usually performed by a helpdesk operator from the admin console. These are typically day to day functions not viewed as part of the User Admin API such as user unlock, PIN reset etc. This is the Helpdesk API

There is additionally read-only functionality provided by the **Reporting API** that can be used to read the status (eg idle, locked) of user accounts.

PRINCIPLES

Following the principles behind the existing Agent API used for authorisation, both of the API?s 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 PINsafe via HTTP and PINsafe 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 PINsafe log. Only authorised PINsafe 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.

In order to use the Admin API, you must create a PINsafe Agent for the computer, or sub-net of computers, that will execute AdminAPI commands. Only the computer(s) defined by this Agent can create, read, update or delete users belonging to this Agent, and the Agent cannot read, update or delete users created by other repositories. The Agent can therefore be regarded as another type of repository. It is possible, but not recommended, to manipulate users created by a normal (e.g. XML or Active Directory) repository, by giving the Agent exactly the same name as the repository. Be aware if you do this, however, that any subsequent User Sync of the repository will potentially overwrite modifications made by the Agent. At present, any PINsafe Agent can act as an AdminAPI repository, but future versions of PINsafe will require you to explicitly enable the "Act as Repository" property of an Agent.

These restrictions do not apply to HelpdeskAPI Agents. Although only Agents can use these commands, it is possible to specify that the HelpdeskAPI command operates on a different repository, or on all repositories.

Sending an Admin API request

An Admin API request must be sent as an HTTP(S) request to the pinsafe application, either as a GET or POST request.

For a GET request, the format is

```
https://[swivelserver]:8080/pinsafe/AdminXML?xml=[request body]
```

For a POST request, the URL is

```
https://[swivelserver]:8080/pinsafe/AdminXML
```

and the XML request should form the content of the POST.

REPORTING API

See Also: Reporting Using Agent-XML How to Guide

Although listed here as a separate API, the Reporting API is actually part of the User Admin API and these elements are only valid as part of an AdminRequest eg

```
<?xml version="1.0" ?>
<AdminRequest secret="MyHelpdeskAgent" version="3.4">
 <Report repository="local">
   <Disabled/>
 </Report>
</AdminRequest>
```

Disabled

Report disabled user accounts. This returns a list of all users from a given repository that are marked as disabled in the PINsafe database. **Note** that this interrogates the PINsafe database and not the repository.

```
 <Report repository="local">
   <Disabled/>
 </Report>
```

Idle

Report users idle since the specified date.

```
 <Report>
   <Idle repository="myRepository" since="12-Mar-2007"/>
 </Report>
```

Note: This does not include users that have been created, but have never used PINsafe.

Locked

Report locked user accounts.

```
 <Report repository=?*?>
   <Locked/>
 </Report>
```

All elements support the optional repository attribute and ?*? is taken to mean ?all repositories.? Therefore the above example will instruct PINsafe to return a list of ALL locked users.

Count

(Since 3.8)

Rather than list users you can request a count of users: For example

```
<AdminRequest secret="secret" version="3.8">
<Report repository="*">
<CountUsers/>
</Report>
</AdminRequest>
```

License information

Check the license details (expiricy and number of users)

```
<SASRequest>
<Version>3.8</Version>
<Secret>secret</Secret>
<Action>GetLicenceInfo</Action>
<LicenceNumber>your-license-key</LicenceNumber>
</SASRequest>
```

AllUsers

(Since 3.8)

Report all users in a repository.

```
<Report repository="*">
 <AllUsers/>
</Report>
```

AllUsersDetailed

(Since 3.9)

Report details of all users in a repository.

```
<Report repository="*">
 <AllUsersDetailed/>
</Report>
```

Responses

The responses to reporting requests are a list of users that meet the definition of the request.

For example

```
<?xml version="1.0"?>
<AdminResponse>
<Report repository="fred">
<Locked>
<User name="jim1"/>
</Locked>
</Report>
</AdminResponse>
```

The exception to this is the idle report as this also includes for each user the time that they last authenticated

```
<?xml version="1.0"?>
<AdminResponse>
<Report repository="*">
<Idle>
<User name="test" lastLogin="2009-06-03 10:38:46.648"/>
<User name="test2" lastLogin="2009-06-03 16:45:19.413"/>
</Idle>
</Report>
</AdminResponse>
```

Response to the Count report is the number of users.

```
<?xml version="1.0"?>
<AdminResponse>
<Report repository="*">
```

```
<CountUsers>
<total>53</total>
<licensed>101</licensed>
</CountUsers>
</Report>
</AdminResponse>
```

Note: The <licensed> element in the Count report response only appears from 3.9.2 onwards.

General Errors and parsing errors are described in Admin API.