AlwaysUp Web Service API

Version 15.0

0. Version History

Version	***Date***	***Updates***
15.0	2024-01-12	Reject all attempts to reboot the computer when that functionality is disallowed. Return the AlwaysUp application configuration in verbose API requests.
14.7	2023-09-05	Security improvements.
14.1	2023-01-26	Fix for cookies
13.3	2022-06-04	Fix for cookies
13.2	2022-01-17	Introduced support for reverse proxies Support starting & restarting in a given session
12.7	2021-03-29	Wait longer for services that take a while to stop
11.0	2018-05-24	Added support for tags Added support for “Waiting” states
8.8	2014-08-18	Added the verbose option for get-status
1.5	2007-12-30	Added reboot; Handle errors consistently; Output file
1.0	2007-09-09	Added restart; provided links for MD5
0.1	2007-08-27	Initial draft; get-status, start and stop operations

1. Overview

The AlwaysUp Web Service API enables developers to control AlwaysUp applications from their own programs. It is built atop generic HTTP/S and XML technologies.

2. Operations

A handful of operations are available to control AlwaysUp applications via AlwaysUp Web Service.

2.1. Common Topics

2.1.1. Authentication

Each operation requires authentication, achieved by sending the MD5 hash of the password used by AlwaysUp Web Service as a parameter on each request. MD5 is discussed in Appendix I.

Note: The current authentication scheme will prevent an eavesdropper from discovering the Web Service password after capturing a request, but it will not defend against him replaying the request verbatim to “impersonate” a valid user. Subsequent versions of this API will address this shortcoming, perhaps hashing with the GMT date to prevent “stale” requests from being processed.

2.1.2. Error Handling

Calling any API operation can result in an error. A return code of 400 indicates an error, and in those instances the HTML body will contain an XML document with one or more descriptive messages. The XML has this format:

```
<?xml version="1.0" encoding="utf-8"?>
<alwaysup-errors>
        <error>Description of the first error</error>
        <error>Description of the second error, etc.</error>
</alwaysup-errors >
```

There can be an unlimited number of error tags, one for each problem, but typically there will only be one. For example, this is the result of a failure to specify a password when performing an operation:

```
<?xml version="1.0" encoding="utf-8"?>
< alwaysup-errors>
        <error>No password was specified</error>
</ alwaysup-errors >
```


2.2. Get Application Status

Synopsis	Returns status on one or all applications
Method	GET
URI	/api/get-status
Parameters	**password** *(mandatory)*: The MD5 hash of the password used by AlwaysUp Web Service (see section 2.1.1). **application** *(optional)*: The name of an AlwaysUp application. **tag** *(optional)*: A tag associated with one or more AlwaysUp applications. **verbose** *(optional)*: If 1 or true, return additional information for the application(s). Details follow. -
Return	On success, the return code is 200 and the body contains an XML document describing the results. The schema for the XML is presented in Appendix II. If neither **application** nor **tag** is provided, status on all applications is returned. On failure, the return code is 400 and XML describing the error(s) is returned in the body (see section 2.1.2).

When the verbose parameter is provided, the following information may also be returned:

- The name of the service created by AlwaysUp.
- The command line used to launch the application. This is the “Application & Arguments” seen in AlwaysUp.
- The entire application configuration, specifying how your application is set up in AlwaysUp. The information will be contained in a “configuration” tag; this XSD describes the contents of that tag: https://www.coretechnologies.com/products/AlwaysUp/AlwaysUpCLT/alwaysup_installservice.xsd
- The identifier of the process being monitored by AlwaysUp
- The ID of the session where the application is running
- The percentage of the CPU being used by the application
- The amount of memory being used by the application (in MB)
- The name of the image being monitored by AlwaysUp.

- The time at which the application was started, in the format YYYY/MM/DD HH:MM:SS
- The version number of the AlwaysUp Web Service executable file.
- The set of tags associated with the application. Each value is separated by a semi-colon (;).

Note: Because of the level of permissions required in gathering this information, some of the fields may not be available to AlwaysUp Web Service. Whenever an item is unknown, its corresponding tag will not be present in the message returned.

Examples:

1. To return the status of the AlwaysUp application called “spider” on computer “myhost” with Web Service password “test”, send:

http://myhost:8585/api/get-status?password=098f6bcd4621d373cade4e832627b4f6&application=spider

If successful, this will return HTTP code 200 and an XML document resembling:

```
<?xml version="1.0" encoding="utf-8"?>
<alwaysup-get-status-response>
	<applications>
		<application>
			<name>spider</name>
			<state>Running</state>
		</application>
	</applications>
</alwaysup-get-status-response>
```

2. To return status of all AlwaysUp applications on computer “myhost” with Web Service password “test”, send:

http://myhost:8585/api/get-status?password=098f6bcd4621d373cade4e832627b4f6

If successful, this will return HTTP code 200 and an XML document resembling:

```
<?xml version="1.0" encoding="utf-8"?>
<alwaysup-get-status-response>
	<applications>
		<application>
			<name>spider</name>
			<state>Running</state>
```



```
                </application>
                <application>
                        <name>FTPServer</name>
                        <state>Stopped</state>
                </application>
        </applications>
</alwaysup-get-status-response>
```


2.3. Start Application

Synopsis	Starts a given application
Method	GET
URI	/api/start
Parameters	**password** *(mandatory)*: The MD5 hash of the password used by AlwaysUp Web Service (see section 2.1.1). **application** *(mandatory)*: The name of the AlwaysUp application. **session-id** (optional): The ID of the session where the application should be started. If not provided, the application is started in Session 0.
Return	On success, the return code is 200. On failure, the return code is 400 and XML describing the error(s) is returned in the body (see section 2.1.2).

Examples:

1. To start the AlwaysUp application named “spider” on computer “myhost” with Web Service password “test”, use:

http://myhost:8585/api/start?password=098f6bcd4621d373cade4e832627b4f6&application=spider

2.4. Stop Application

Synopsis	Stops a given application
Method	GET
URI	/api/stop
Parameters	**password** *(mandatory)*: The MD5 hash of the password used by AlwaysUp Web Service (see section 2.1.1). **application** *(mandatory)*: The name of the AlwaysUp application.
Return	On success, the return code is 200. On failure, the return code is 400 and XML describing the error(s) is returned in the body (see section 2.1.2).

Examples:

1. To stop the AlwaysUp application named “spider” on computer “myhost” with Web Service password “test”, use:

http://myhost:8585/api/stop?password=098f6bcd4621d373cade4e832627b4f6&application=spider

2.5. Restart Application

Synopsis	Restarts a given application (or starts it if it is not running)
Method	GET
URI	/api/restart
Parameters	**password** *(mandatory)*: The MD5 hash of the password used by AlwaysUp Web Service (see section 2.1.1). **application** *(mandatory)*: The name of the AlwaysUp application. **session-id** (optional): The ID of the session where the application should be started. If not provided, the application is started in Session 0.
Return	On success, the return code is 200. On failure, the return code is 400 and XML describing the error(s) is returned in the body (see section 2.1.2).

Examples:

1. To restart the AlwaysUp application named “spider” on computer “myhost” with Web Service password “test”, use:

http://myhost:8585/api/restart?password=098f6bcd4621d373cade4e832627b4f6&application=spider

2.6. Get Application Output File

Synopsis	Returns the output file associated with a given application
Method	GET
URI	/api/output_file
Parameters	**password** *(mandatory)*: The MD5 hash of the password used by AlwaysUp Web Service (see section 2.1.1). **application** *(mandatory)*: The name of the AlwaysUp application.
Return	On success, the return code is 200 and the contents of the output file are returned in the body. On failure the return code is 400 and XML describing the error(s) is returned in the body (see section 2.1.2). If output is not being captured for the application, the return code is 400 and XML with the error *"Output is not being captured for application <app name>"* is returned in the body (see section 2.1.2).

Examples:

1. To fetch the output file for the AlwaysUp application named “spider” on computer “myhost” with Web Service password “test”, use:

http://myhost:8585/api/output_file?password=098f6bcd4621d373cade4e832627b4f6&application=spider

Note that an error will be returned if no output is being captured for “spider”.

2.7. Report Application

Synopsis	Returns an XML report on the application's activity over the previous 7 days.
Method	GET
URI	/api/report
Parameters	**password** *(mandatory)*: The MD5 hash of the password used by AlwaysUp Web Service (see section 2.1.1). **application** *(mandatory)*: The name of the AlwaysUp application.
Return	On success, the return code is 200 and the body contains the XML report. The schema for the XML is presented in Appendix III. On failure, the return code is 400 and XML describing the error(s) is returned in the body (see section 2.1.2).

Examples:

1. To get a report for the AlwaysUp application named "spider" on computer "myhost" with Web Service password "test", use:

http://myhost:8585/api/report?password=098f6bcd4621d373cade4e832627b4f6&application=spider

2.8. Restart/Reboot Computer

Synopsis	Restarts the computer
Method	GET
URI	/api/reboot
Parameters	**password** *(mandatory)*: The MD5 hash of the password used by AlwaysUp Web Service (see section 2.1.1). **immediate** *(optional)*: true to restart the computer immediately, or false to first notify all interactive users of the shutdown (and wait for up to 60 seconds doing that) prior to rebooting. If not specified, the effect is the same as specifying false. When true, all AlwaysUp applications will be stopped before the reboot is initiated.
Return	On success, the return code is 200. On failure, the return code is 400 and XML describing the error(s) is returned in the body (see section 2.1.2).

Examples:

1. To restart computer “myhost” with Web Service password “test”, use:

   http://myhost:8585/api/reboot?password=098f6bcd4621d373cade4e832627b4f6

As part of the restart procedure, Windows will notify all users logged in of the restart and wait up to 60 seconds for them to save their work and log off.

Note that AlwaysUp Web Service can not cancel a restart once the request has been made, but a user with sufficient administrative rights logged in to the PC may be able to interrupt the process.

Also note that when **immediate** is false or is not specified, AlwaysUp Web Service will not explicitly stop any application prior to rebooting and simply invokes the “normal” windows shutdown procedure (which should gracefully stop each application and run any custom stop commands configured).

3. Troubleshooting and Reporting Problems

Please consult the AlwaysUp FAQ for troubleshooting tips and answers to frequently asked questions:

https://www.coretechnologies.com/products/AlwaysUp/AlwaysUp_FAQ.html

If you encounter a problem while using AlwaysUp, please send email to:
support@CoreTechnologies.com

Be sure to include the following information:

- Your Operating System
- The version of AlwaysUp Web Service in use
- The version of AlwaysUp in use
- Detailed steps for reproducing any software bugs/issues

Feel free to send requests for enhancements to the same address.

4. Appendix I: The MD5 Algorithm

MD5 is a popular and widely used one-way hash function. Given arbitrary text, MD5 will produce a unique 128-bit value, from which it is nearly impossible to derive the original text (hence the algorithm's one-way designation). Find out more here:

http://en.wikipedia.org/wiki/MD5

For testing purposes, these tools will generate a MD5 hash from any given text:

http://www.md5hashgenerator.com/
http://www.miraclesalad.com/webtools/md5.php

5. Appendix II: XML Schema for the result of the Get Application Status operation

```
<?xml version="1.0"?>
<xs:schema id="alwaysup-get-status-response"
targetNamespace="http://tempuri.org/XMLFile1.xsd"
xmlns:mstns="http://tempuri.org/XMLFile1.xsd"
xmlns="http://tempuri.org/XMLFile1.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata" attributeFormDefault="qualified"
elementFormDefault="qualified">
  <xs:element name="alwaysup-get-status-response"
msdata:IsDataSet="true" msdata:EnforceConstraints="False">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="applications" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="application" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="name" type="nonEmptyStringType"
minOccurs="1" maxOccurs="1" />
                    <xs:element name="state" type="stateType"
minOccurs="1" maxOccurs="1" />
                    <xs:element name="service-name"
type="nonEmptyStringType" minOccurs="0" maxOccurs="1" />
                    <xs:element name="command-line"
type="nonEmptyStringType" minOccurs="0" maxOccurs="1" />
                    <xs:element name="tags" type="nonEmptyStringType"
minOccurs="0" maxOccurs="1" />
                    <xs:element name="pid" type="xs:positiveInteger"
minOccurs="0" maxOccurs="1" />
                    <xs:element name="session-id"
type="xs:positiveInteger" minOccurs="0" maxOccurs="1" />
                    <xs:element name="cpu-percent"
type="xs:positiveInteger" minOccurs="0" maxOccurs="1" />
                    <xs:element name="memory-mb-used" type="xs:decimal"
minOccurs="0" maxOccurs="1" />
                    <xs:element name="image-name"
type="nonEmptyStringType" minOccurs="0" maxOccurs="1" />
                    <xs:element name="start-time"
type="nonEmptyStringType" minOccurs="0" maxOccurs="1" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
```



```
  </xs:element>
  <!-- ++++++++++++ Referenced types ++++++++++++ -->
  <!-- nonEmptyStringType:  represents a non-empty string. -->
  <xs:simpleType name="nonEmptyStringType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- stateType: Valid states for an application. -->
  <xs:simpleType name="stateType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Running"/>
      <xs:enumeration value="Stopped"/>
      <xs:enumeration value="StartPending"/>
      <xs:enumeration value="StopPending"/>
      <xs:enumeration value="Paused"/>
      <xs:enumeration value="PausePending"/>
      <xs:enumeration value="ContinuePending"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Note that the valid states are Running, Stopped, StartPending (the application is starting), StopPending (the application is stopping), Paused, PausePending (the application is being paused), and ContinuePending (the application is resuming from a pause). AlwaysUp does not allow you to manipulate the Pause-related states, but you can do so from the Windows Service Control Manager.

The schema file is also available at:
https://www.coretechnologies.com/products/AlwaysUp/AlwaysUpWebService/alwaysup-get-status-response.xsd

6. Appendix III: XML Schema for the result of the Report Application operation

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="alwaysup-report">
    <xs:complexType>
      <xs:sequence>
        <!-- General information -->
        <!-- The name of the application. -->
        <xs:element name="application-name" type="nonEmptyStringType"
/>
        <!-- The date that the report was generated. -->
        <xs:element name="date" type="dateTimeStringType" />
        <!-- Information on the computer. -->
        <xs:element name="computer">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded"
name="name" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <!-- Details for each day -->
        <xs:element name="days">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="day" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <!-- Summary information for this day -->
                    <xs:element name="summary">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="memory" minOccurs="0"
maxOccurs="1">
                            <xs:complexType>
                              <xs:attribute name="average"
type="xs:float" use="optional" />
                              <xs:attribute name="peak" type="xs:float"
use="optional" />
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="cpu" minOccurs="0"
maxOccurs="1">
                            <xs:complexType>
                              <xs:attribute name="average"
type="xs:float" use="optional" />
                              <xs:attribute name="peak" type="xs:float"
use="optional" />
                            </xs:complexType>
```



```
                        </xs:element>
                      </xs:sequence>
                      <xs:attribute name="num-times-run"
type="xs:nonNegativeInteger" use="optional" />
                      <xs:attribute name="num-restarts"
type="xs:nonNegativeInteger" use="optional" />
                      <xs:attribute name="time-running"
type="xs:nonNegativeInteger" use="optional" />
                      <xs:attribute name="availability"
type="xs:float" use="optional" />
                    </xs:complexType>
                  </xs:element>
                  <!-- Detailed information for this day -->
                  <xs:element name="details">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element minOccurs="0"
maxOccurs="unbounded" name="detail">
                          <xs:complexType>
                            <xs:simpleContent>
                              <xs:extension base="xs:string">
                                <xs:attribute name="time"
type="dateTimeStringType" use="required" />
                              </xs:extension>
                            </xs:simpleContent>
                          </xs:complexType>
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                <xs:attribute name="date" type="dateTimeStringType"
use="required" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- ++++++++++++ Referenced types ++++++++++++ -->
<!-- nonEmptyStringType:  represents a non-empty string. -->
<xs:simpleType name="nonEmptyStringType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
  </xs:restriction>
</xs:simpleType>
<!-- dateTimeStringType:  represents a date string, in the 19-digit
format: YYYY/MM/DD HH:MM:SS. -->
<xs:simpleType name="dateTimeStringType">
  <xs:restriction base="xs:string">
    <xs:minLength value="19"/>
    <xs:maxLength value="19"/>
  </xs:restriction>
</xs:simpleType>
```



```
</xs:schema>
```

The schema file is also available at:
https://www.coretechnologies.com/products/AlwaysUp/AlwaysUpWebService/alwaysup-report-response.xsd

