Updated May 31st, 2018

Overview

The OpenControl protocol supports third-party integration with Solstice hosts through a simple RESTful API. OpenControl can be an important component in a Solstice deployment because it allows integrators and installers to more deeply integrate Solstice with existing room infrastructure. For example, it can be used to monitor the status of a Solstice host, capture usage statistics, and configure endpoints. In short, OpenControl exposes much of the functionality found in the Solstice Dashboard through a straightforward communications protocol can that can be read/written both by humans and machines. Creative uses of the API include automatically emailing IT administrators when a Pod's new settings do not meet enterprise security standards and enabling dynamic switching between input sources for digital signage.
OpenControl uses HTTP GET and POST commands to receive and send data to a Solstice host. The approach ensures that the API is composed of simple transactions that can be sent to a Pod without requiring complex management of state by a 3rd party developer. Any language, on any device, that can issue HTTP GET and POST commands to the appropriate URLs of the Solstice host, then, can be integrated into a Solstice deployment.

Quick Links:

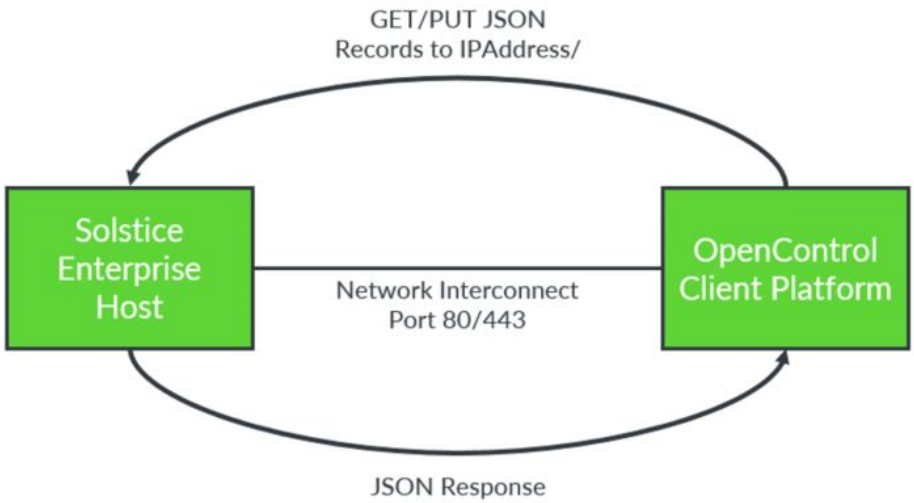
What is Required to use the OpenControl API?

- A Solstice Pod or Solstice Windows Software host with an Enterprise Edition license. Note that the Pod and Windows host respond differently to different commands and that some commands are Pod or Windows only; see command list for details.
- A client platform for issuing commands to the Solstice host.
- A network connection between the client and Solstice host.

Example Uses of the OpenControl API

- Capturing statistical usage data periodically and storing that data in a third-party database for analytics.
- Integrating configuration options into a third-party IT dashboard suite, allowing IT dashboards to incorporate status information about a Solstice deployment.
- Clearing the screen of posted content or booting users from a Solstice session from in-room control panels.
- Displaying information about a Solstice session – for example the Session Key – on a control panel or second display.
- Updating the Solstice host splash-screen messaging from a third-party application.

The OpenControl communications protocol utilizes JSON structures to exchange information between the Solstice host and any number of clients. The protocol uses port 80 or 443. Solstice hosts receive request records and respond to requests by either carrying out an action (i.e. modifying configuration, performing an action) or by responding with a JSON response. OpenControl clients can only communicate with Enterprise Edition Solstice hosts. Standard Solstice Pods (i.e. non-Enterprise Edition) will need to be upgraded to Enterprise Edition before using the OpenControl API. The figure below depicts the main components needed to implement OpenControl.



OpenControl operates over existing TCP/IP network infrastructure to allow third-party client platforms to control and query Solstice endpoints. The protocol is based on a REST architecture that encapsulates communication into independent GET/POST events, sent to particular URLs on the Solstice Pods, each of which result in a JSON formatted response.

Back to Top

Using the OpenControl API

The protocol is divided into areas based on functional types. These types are: Config, Stats, and Command. Each is associated with a different URL. By sending an appropriately formatted JSON record to the corresponding URL, third-party developers can query and set various values/settings related to the Solstice host. These URLs are:

- /api/config – Used for posting and getting information related to administrative configuration of Solstice host. For example, modifying the network settings of a Pod.
- /api/stats – Used to get instantaneous status about a Solstice host. For example, capturing the number of users currently connected to a Solstice Windows Software endpoint.
- /api/control – Used to post commands to a Solstice host that will impact runtime behavior. For example, clearing a screen of all media.
- Other URLs are used in the Calendar and Version APIs; see their sections below for details.

Users of OpenControl can set values using a POST structure to the appropriate URL. This is used to modify configuration, or POST actions to a Solstice host to control runtime behavior. In addition, users can query current configuration and status using GET commands to the appropriate URL.

Setup

This section is intended for users who want to demo the API capability or run one of the Mersive supplied Python applications, and assumes no familiarity with general purpose programing. If you are familiar with programming and have a preferred way of sending and receiving HTTP GET and POST commands, jump to the list of commands.

If you simply want to experiment with receiving and sending information via the API, Postman is a great tool that can help you develop an intuitive grasp of the capabilities and workflows of the OpenControl API. Enter the URL of your target host or server and click 'GET' to confirm connectivity with your target and see the results of every key:value set associated with that target.

If you want to run complete scripts and are starting from scratch, follow the comprehensive instructions below.

The following setup steps assume a Windows operating system on the client device. If you have a different operating system, you will need to modify the steps to suit your platform. It also assumes that you want to use Python version 2.7 to communicate with the API. While the OpenControl API is agnostic to client programming language and version, the examples in this document utilize Python 2.7.

1. Download Python 2.7 from python.org/downloads/. Python 3.6 is not fully compatible with 2.7 and the example scripts will not run properly if you are using Python 3.6.
2. Install as close to the main drive as possible. The default location is likely C:\Python27, which is perfect.
3. Download get-pip from https://pip.pypa.io/en/stable/installing/ and move the file into the same folder as your new installation of Python 2.7.
4. Open Windows PowerShell.
5. Navigate to the Python directory. If you installed in the default location, type:

   cd C:\Python27

   If you installed into a different directory, navigate through your file system using "cd [folder name]" to enter a subfolder or "cd .. " to move up a level. The command "ls" (that's a lowercase L) lets you see everything inside the current directory.
6. Once you're in the Python folder, type "ls" to verify that get-pip.py is in the folder.
7. If so, type "py get-pip.py" and wait for pip to install.
8. Next, install the requests package with the following command:

   python -m pip install requests

   Wait for requests to finish installing. Information on this package is available at http://docs.python-requests.org/en/master/.

Once the requests package is installed, you can GET/POST individual messages to or from specific IP addresses directly from the Python terminal window (access by double-clicking the python.exe file), or you can write a Python script in an editor (such as Notepad++) and run the script from PowerShell or the Python terminal window.
Now, test that you have network connectivity to a Solstice IP address (Pod or Software Instance) by pinging the IP address. If the IP address is http://192.168.3.127, the shell command would be:
>>> ping 192.168.3.127
Wait for the ping to finish. The result should be 4 packets sent and 4 packets received.

At this point, we recommend downloading and running the example GET script to see all the available options. Download the script from our GitHub repository and move it to a known location. You can double-click the file to run it immediately, or open it in an editor to modify the script.

If you start writing your own script, you will need to import the **requests** package before attempting to execute any GET/POST code. You will also need to import **sys**, and will likely want **time** as well. Though it seems redundant, you can avoid errors later on by explicitly defining true=1 and false=0 in your new Python window or at the top of your script.
Note: the ">>>" characters are not something you type into the window – they are already at the start of each line in the Python terminal. They are shown in the code here to clarify where new lines begin.

In general, start with all the following commands:

```
>>> import sys
>>> import time
>>> import datetime
>>> import requests
>>> import json
>>>
>>> true=1
>>> false=0
```

JSON Record Structure

POST and GET records sent to and received from the different URLs are made up of key/value pairs that represent the various capabilities exposed through the API. This structure uses the JSON syntax, and, in general, looks like:

{key1: value1, key2: value2, … keyN, valueN}

POST records do not need to contain all key/value pairs and can contain any subset of key/values based on the needs of the integration. The order of a request record is not important, so long as the key/value pairs follow the format above. For example, POSTing the string '{key2:value2}' to the appropriate URL of a Solstice host can set that value.

In some cases, key/value pairs are organized hierarchically based on logical groupings. In these cases, the JSON syntax is simply nested within the value of a particular key. Those key/value pairs appear within brackets and are separated from other key/value pairs with a comma. For example:

{**key1:value1, key2:value2, GroupKey:{ NestedKey1:GroupValue1, …, NestedKeyN:GroupValueN }, key3:value3}**

This represents a JSON structure that contains three key/value pairs at the top level, and a set of key/value pairs that are grouped within 'GroupKey'.
Using this syntax, for example, a user could set the display name for a particular Solstice host by sending the following JSON record:

{m_displayInformation:{m_displayName:'NewDisplayName'}}

Securing API Communications

In order to ensure that only valid third-party users are able to communicate with Solstice Hosts, the administration password, when set, must be provided with each POST or GET record. The administration password, for a particular host, can be set both in the Solstice host Configuration Panel, or through the Solstice Dashboard. **It is important to note that if no password is set, then any third-party application can utilize the OpenControl APIs to modify a Solstice host over the network.**

When an administrator password is set, each POST request record must include a key/value pair that is: 'password: admin_password' at the top level of the record. A request record sent to a Solstice host that has password enabled, then, must follow this format:

{password: admin_password, key1: value1, key2: value2, … keyN, valueN}

In the case of a GET record, the password is simply appended to the GET URL request as follows:

?password=admin_password

Back to Top

Basic GET

There are two URLs for each IP address that will respond with text to a GET command. For our example IP address of 192.168.3.127, the valid URLs are:
http://192.168.3.127/api/stats
http://192.168.3.127/api/config

While there is some overlap in information, the "stats" URL generally gives a snapshot of important values and instantaneous usage while the "config" URL gives more detail about everything from enabled options to display screen layout. Note that while you can GET from either URL, you can only POST to /api/config.

To test the stats URL, set up your environment as described in the previous section and send the following command (with your own URL):
>>> rs=requests.get('http://192.168.3.127/api/stats')

If you have an admin password protecting access to the display, append the password in the following way, with "adminpassword" as whatever the actual password is for the display.
>>> rs=requests.get('http://192.168.3.127/api/stats?password=adminpassword')

To display the raw results, type:
>>> rs.text

You should see a continuous chunk of text with the first distinct value of "m_displayID". To pull out a specific value in a clean format, the string needs to be converted to a dictionary of key:value pairs.
>>> rstats = eval(rs.text)

If you print the new value (rstats.text) you will still see a chunk of continuous text, but the formatting will be slightly different. We can now use get() to find specific terms, like the display name. m_displayName is a part of the m_displayInformation group key, so we will use nested get() commands:
>>> print "Display Name:", rstats.get('m_displayInformation',{}).get('m_displayName')

The result should be "Display Name: Pikes Peak" with the actual display name reflecting the name on your Solstice display. Any of the values can be pulled from the list in this manner.

Basic POST

Many of the key:value pairs that can be read using GET can also be changed through the API using POST. While the same values may appear in the GET of both URLs, POSTs may only be sent to the /api/config URL:
http://192.168.3.127/api/config

The requests command now changes from GET to POST and requires a payload parameter. To change the display name, use the following command:
>>> r=requests.post('http://192.168.3.207/api/config', json={'m_displayInformation':{'m_displayName':'New Name'}})

If you have an admin password protecting the display, send the following command where 'adminpassword' is the actual password for the display:
>>> r=requests.post('http://192.168.3.207/api/config', json={'password: admin_password',
'm_displayInformation':{'m_displayName':'New Name'}})

NOTE: if you copy either of these lines directly into Python or a formatted text editor and get a syntax error, the single quotes are likely to blame. They may come through formatted such that requests.post() does not recognize them. If this happens, manually delete each mark and re-enter it in the terminal window to get the appropriate format.
To see the new name, run GET on the same URL and look for the 'm_displayName' key, which should now have a value of 'New Name':
>>> rc=requests.get('http://192.168.3.127/api/config')
>>> rconfig=eval(rc.text)
>>> print "Display Name:", rconfig.get('m_displayInformation',{}).get('m_displayName')

If you have an admin password protecting the display, you will need to add in the password as before.

Back to Top

GET/POST Example Script (Python 2.7)

The complete script provides additional comments and the option for an admin password.

>Download complete script

This pared down version gets the Display Name from both /config and /stats URLs, changes the name, then gets both values again to confirm the name has changed.

```
import sys
import requests
import random
from random import choice
true=1
```

```
false=0

newname = “New Name”
myurl = “http://192.168.3.227”
admin_password = “”
mystatsurl = myurl+’/api/stats’
myconfigurl = myurl+’/api/config’

rs=requests.get(mystatsurl)
rc=requests.get(myconfigurl)
rstats=eval(rs.text)
rconfig=eval(rc.text)

print “Current Display Name from Stats:”, rstats.get(‘m_displayInformation’,{}).get(‘m_displayName’)
print “Current Display Name from Config:”, rconfig.get(‘m_displayInformation’,{}).get(‘m_displayName’)

r=requests.post(myconfigurl, json={‘password’:admin_password,’m_displayInformation’:{‘m_displayName’:newname}})

print “Changing Name to: “,newname
print “……”

rs=requests.get(mystatsurl)
rc=requests.get(myconfigurl)
rstats=eval(rs.text)
rconfig=eval(rc.text)

print “New Display Name from Stats:”, rstats.get(‘m_displayInformation’,{}).get(‘m_displayName’)
print “New Display Name from Config:”, rconfig.get(‘m_displayInformation’,{}).get(‘m_displayName’)
```

Back to Top

Configuration API

Solstice Host URL: IPAddress/api/config

The configuration API is focused on setting and reading configuration settings that are available in the Configuration Panel and the Solstice dashboard. These are admin settings that are concerned with customization, security, and network configuration options.
The top level keys are primarily associated with device-specific settings and are not grouped into a sub-hierarchy. For each table, other than the ‘Top Level’ table that does not require a hierarchical key, the hierarchical key is shown in the first row. This is the key to the set of values that can be set in that table.

Product/Global Settings

(Top Level)

Key	Type	Get/Post	Description
m_displayId	string	Get	This is the unique identifier that Solstice uses to manage a single instance, regardless of how the display is named or its current IP Address
m_serverVersion	string	Get	The current software version running on the Solstice host
m_productName	string	Get	The name of the Solstice software (Solstice).
m_productVariant	string	Get	The generation and type of Pod hardware as a name. For example, ‘Gen1’ or ‘Gen2’
m_productHardwareVersion	int	Get	The Pod hardware generation as a version number (Pod-only). For example, 1 or 2. Software returns 9999.

Display Communications Settings

m_displayInformation

Key	Type	Get/Post	Description
m_ipv4	string	Get	The current primary IP Address assigned to the Solstice display/endpoint.
m_displayName	string	Get/Post	The display name used for Solstice discovery. This name is shown on the Solstice splash-screen and appears in the client discovery list for connecting to the Solstice display.
m_hostName	string	Get/Post	The Solstice host device's current hostname.
m_port	int	Get/Post	The base port that Solstice will utilize for TCP/IP communications. Solstice uses three ports defined by the base port value, +1, and +2.

Global Display Settings

m_generalCuration

Key	Type	Get/Post	Description
language	string	Get/Post	A code that denotes the current language setting. Valid options are: "en_US" for US English, "ja_JP" for Japanese, "de_DE" for German, "fr_FR" for French, "es_ES" for Spanish, "zh_TW" for Traditional Chinese, and "xx".
showSplashScreen	bool	Get	Show or hide the splash screen background image when the splash-screen is visible.
localConfigEnabled	bool	Get	Enable or disable access to the local Configuration Panel on the Solstice host (via mouse/keyboard). When disabled the onscreen configuration menu is no longer visible to users.
browserConfigEnabled	bool	Get	If this is disabled, the API cannot interact with the host since the device considers the API browser configuration.
autoConnectOnClientLaunch	bool	Get/Post	Enable or disable quick-connect auto launch
hideOnLastClientDisconnect	bool	Get	Windows Only – hide the display software if no connections are active
launchOnClientConnect	bool	Get	Windows Only – bring display software to foreground when a connection is made
launchOnSystemStart	bool	Get	Windows Only – start display software when host machine boots
theme	int	Get	Windows Only – selects between the 6 preset themes. Zero indexed, so the options are 0-5.
advancedRenderingEnabled	bool	Get/Post	Windows Only – enable (1) or disable (0) advanced rendering for better animations at the expense of host processing cycles.
windowMode	int	Get	Windows Only – shows whether Solstice is displayed as window app (0), fixed size (1) , or fullscreen (2)
windowTop	int	Get	Windows Only – if windowMode = 1, this is the top coordinate for the fixed window
windowLeft	int	Get	Windows Only – if windowMode = 1, this is the left coordinate for the fixed window.
windowWidth	int	Get	Windows Only – if windowMode = 1, this is the width in pixels of the fixed window.
windowHeight	int	Get	Windows Only – if windowMode = 1, this is the height in pixels of the fixed window.

Authentication Modes

m_authenticationCuration

Key	Type	Get/Post	Description
authenticationMode	int	Get	Pre 3.0 Solstice only – security on host is open (0), screen key (1), password (2), moderated (3), or select at runtime (4)
screenKeyEnabled	bool	Get/Post	Check whether screen is key protected
moderatorApprovalDisabled	bool	Get/Post	Check whether moderator mode is enabled or disallowed
sessionKey	string	Get	The current session key currently displayed on the screen and required to be entered by clients before connecting to the display

Back to Top

Main Network and Feature Record

m_networkCuration

Key	Type	Get/Post	Description
connectionShowFlags	Bitwise flags (post as decimal int)	Get/Post	Bitwise flags for showing various connection related information on the Solstice splash-screen. Pod only. Windows Software returns 32 '1's. Bit 1: Main Screen – Display Name Enabled Bit 2: (Don't Care) Bit 3: Show IP Address on Main Screen Bit 4: Presence Bar – Display Name Bit 5: Presence Bar – IP Address Bit 6: Show Screen Key on Main Screen Bit 7: Presence Bar – Screen Key Bit 8: Show Presence Bar Bit 9: Connect by App Instructions Bit 10: App Instructions – Artwork/Icons Bit 11: Connect by Web Instructions Bit 12: Web Instructions – Artwork/Icons Bit 13: Show SSID Info (when enabled)
discoveryBroadcastEnabled	bool	Get/Post	Enable or Disable UDP Discovery broadcast. When enabled the Solstice host will broadcast discovery information on local network every 5 seconds allowing clients to discover and connect.
publishToNameServer	bool	Get/Post	Publish discovery information to Solstice Discovery Service so clients can discover the Solstice host without the need for broadcast traffic/discovery.
maximumConnections	int	Get/Post	The current number of maximum allowable simultaneous connections. Cannot be set past maximum allowable based on license for host.
maximumLicensedConnections	int	Get	Number of maximum allowable simultaneous connections based on the installed license.
maximumImageSize	int	Get/Post	Maximum size of an image, in bytes, shared by clients. Clients that share images past the maximum will automatically be resized (to save resources).

maximumPublished	int	Get/Post	Maximum number of posts allowed. Users who post past this limit will be given a message that the system is busy.
maximumAirPlayUsers	int	Get/Post	Maximum number of simultaneous AirPlay users allowed (ie iOS mirroring posts). Cannot be set higher than 4.
sdsHostName	string	Get/Post	Primary hostname or IP Address of Solstice Discovery Service to list discovery information.
sdsHostName2	string	Get/Post	Secondary hostname or IP Address of Solstice Discovery Service to list discovery information.
remoteViewMode	int	Get/Post	Enable or disable Browser Look-In feature. 0=disabled, 1=enabled, 2=allow users to toggle on/off at runtime. Passing a value other than 0,1, or 2 disables look-in.
firewallMode	int	Get/Post	Modifies Dual-Network firewall (Pod only) to enable or disable internet traffic between two network interfaces. 0=Block all traffic, 1=Allow ports 80/443.
postTypeDesktopSupported	bool	Get/Post	Enable or disable support for PC/desktop full screen sharing.
postTypeApplicationWindowSupported	bool	Get/Post	Enable or disable support for application window sharing.
postTypeMediaFilesSupported	bool	Get/Post	Enable or disable support for image and video file sharing.
postTypeAirPlaySupported	bool	Get/Post	Enable or disable support for iOS mirroring.
postTypeAndroidMirroringSupported	bool	Get/Post	Enable or disable support for Android full screen mirroring.
bonjourProxyEnabled	bool	Get/Post	Enable or disable the Bonjour Proxy feature that allows iOS users to discover display to mirror to via the Solstice App instead of the Bonjour protocol.
ethernetEnabled	bool	Get/Post	Enable or disable the Ethernet network adapter (Pod-only).
ethernetGatewayCheckEnabled	bool	Get/Post	Enable or disable the gateway check (Pod-only).
wifiMode	int	Get	Set the wireless network adaptor mode 0 = Off, 1 = Client Mode/Attach to existing wireless, 2= Wireless Access Point.
bulletinEnabled	bool	Get/Post	Enable or disable bulletin text across top of Solstice-enabled Display
bulletinText	string	Get/Post	Text to be displayed in bulletin
emergencyEnabled	bool	Get/Post	Enable or disable the emergency broadcast
emergencyText	bool	Get/Post	Text of emergency broadcast
wifiConfig (Pod Only)			
ssid	string	Get	SSID of the host wireless network to connect to via wireless client mode.
security	int	Get	Security protocol of the Pod wireless network (Pod-only). 0=open, 1=WEP, 2=WPA, 3=WPA2, 4=EAP
eap	int	Get	EAP method of authentication when in EAP mode (Pod-only). 0=None, 1=PEAP, 2=TLS, 3=TTLS, 4=PWD, 5=SIM, 6=SIM
phase2	int	Get	Phase2 authentication method. 0=None, 1=PAP, 2=MSCHAP, 3=MSCHAPv2, 4=GTC.

password	string	Get	Password used to authenticate to host network. Get returns "*"
dhcp	bool	Get	Enable or disable DHCP protocol on wireless interface.
staticIP	string	Get	Static IP address to assign to device on wireless interface.
gateway	string	Get	Wireless interface gateway.
prefixlength	int	Get	Prefix setting for wireless interface.
dns1	string	Get	First DNS for wireless interface.
dns2	string	Get	Second DNS for wireless interface.
apConfig (Pod Only)			
SSID	string	Get	SSID for standalone wireless access point .
SecurityMode	int	Get	Security protocol for wireless interface when in WAP mode (Pod-only). 0=open, 3=WPA2.
PSK	string	Get	Password to use for authenticating users connecting to wireless access point.
ethernet (Pod Only)			
dhcp	bool	Get	Enable or disable DHCP for the Ethernet network interface.
staticIP	string	Get	Static IP to assign the device's Ethernet network interface (Pod-only).
gateway	string	Get	Gateway IP Address for the Ethernet network interface (Pod-only).
prefixLength	int	Get	Prefix length value (netmask, Pod-only).
dns1	string	Get	First DNS Server IP Address (Pod-only).
dns2	string	Get	Second DNS Server IP Address (Pod-only).
httpProxyServerSettings			
enabled	bool	Get	Enable or disable use of an HTTP Proxy.
ip	string	Get	IP Address of HTTP proxy server.
port	int	Get	Communications port of HTTP proxy server.
username	string	Get	Username to be used when authenticating to the HTTP proxy server.
password	string	Get	Password to be used when authenticating to the HTTP proxy server.
httpsProxyServerSettings			
enabled	bool	Get	Enable or disable use of an HTTPS Proxy.
ip	string	Get	IP Address of HTTPS proxy server.
port	int	Get	Communications port of HTTPS proxy server.
username	string	Get	Username to be used when authenticating to the HTTPS proxy server.
password	string	Get	Password to be used when authenticating to the HTTPS proxy server.

m_networkCuration also has a nested group key 'm_rssFeedList' that can GET or POST an array of messages to be displayed in the RSS feed across the top of a Solstice-enabled display. It may include custom messages or standard RSS urls. For example:

```
"m_rssFeedList": [
    {
    "enabled": true,
    "name": "Custom Message",
    "length": 0,
    "uri": "This is a test123"
    },
```

```
        {
        “enabled”: true,
        “name”: “solstice wireless display”,
        “length”: 3,
        “uri”: “https://www.mersive.com/go.xml”
        }
]
```


Back to Top

Licensing

m_licenseCuration

Key	Type	Get/Post	Description
licenseStatus	int	Get	0=No license; 1=Error reading license; 2=License OK; 3=License Expired
trustFlags	int	Get	7=fully trusted; <7 and the license isn’t trusted and must be repaired
fulfillmentType	string	Get	“PUBLISHER ACTIVATION” or “TRIAL”
enabled	bool	Get	Tells whether the license on the machine is enabled (1) or disabled (0)
fulfillmentId	string	Get	A unique numerical ID used to distinguish between different machines
entitlementId	string	Get	activation code used to activate the license on this machine
productId	string	Get	“Solstice”
suiteId	string	Get	Should be blank. Internal use.
expirationDate	string	Get	“permanent” or the date the license will expire
featureLine	string	Get	The license string returned from the license server
numDaysToExpiration	int	Get	999999999 if the license is permanent, otherwise the number of days until it expires
maxUsers	string	Get	max number of users allowed by license. eg. “Unlimited”, “4”
licensing_maxPosts	int	Get	maximum number of posts allowed by the license type.
licensing_maxPostsIsConfigurable	bool	Get	Is the user allowed to change the value of maxPosts?
licensing_atMaxPostsReplace	bool	Get	When the max post count is reached, should the next post replace (1) or not display (0)?
licensing_maxUsers	int	Get	maximum users allowed (0=Unlimited)
licensing_maxUsersIsConfigurable	bool	Get	Is the user allowed to change the value of maxUsers?
licensing_remoteViewEnabled	bool	Get	Can remote view be enabled?
licensing_remoteViewIsConfigurable	bool	Get	Is remote view able to be configured?
licensing_runtimeAccessControls	bool	Get	Can users can select the runtime access (moderated, screen key, etc.)?

Passwords

m_userGroupCuration

Key	Type	Get/Post	Description
adminPassword	string	Post	Administrative password for host. Note – Get on this value will always return “unknown”.
passwordValidationEnabled	bool	Get/Post	Enable/Disable validation of admin password using the following rules:

- Minimum of 8 characters.
- At least one uppercase and one lowercase character.
- At least one number or special character.

System Settings Record

m_systemCuration

Key	Type	Get/Post	Description
autoDateTime	bool	Get/Post	Enable or disable the use of an NTP time server. If disabled, date and time is set manually.
ntpServer	string	Get/Post	Non-default NTP time server IP Address. If left blank, a default internet time server will be used when autoDateTime is true.
dateTime	int	Get	Date and time value, represented as a 64-bit integer in milliseconds since January 1, 1970, 00:00:00 GMT.
timeZone	string	Get/Post	Time zone code represented as a string from the set of available time zones.
timeZones	String array	Get	Array of available time zone strings. See Valid Time Zones.

Splashscreen Commands

The background image shows on a Solstice-enabled display when no content is shared. The 'classic' splash screen has a single, static image, while the 'modern' splash screen can have up to 6 custom images in the carousel.

Changing the Classic Splash Screen Image:
POST your desired file (key is "file") to IPaddress/api/config/splashbackground as type formData. In Postman or a similar tool, you can do this by entering the URL, selecting 'POST', entering 'file' as the key, and changing the value type from 'text' to 'file'. This will allow you to upload a file from your computer and post it to the Solstice host.
Reset the background to the default image by changing the value type back to 'text' and entering 'reset default' as the value. POST to the URL to see the change on your Solstice host.

Changing the Modern Splash Screen Image(s):
There are 6 slots for custom images in the modern splash screen carousel, numbered 0-5. To upload a new image or reset a specific image to the carousel, use the same command as the classic splash screen appended with '/n' where n is the number of the image to be changed.

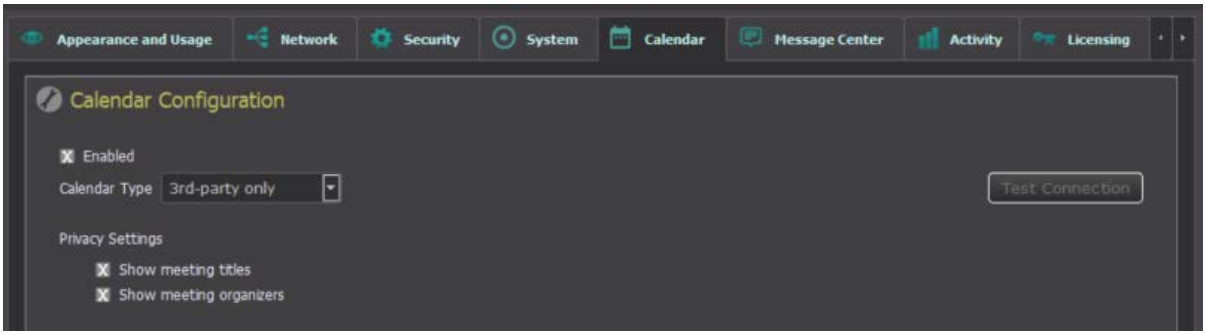
For example, to upload a new image to the 4th place in the carousel, you would POST your image file to IPaddress/api/config/splashbackground/3.

Note that you cannot remove images from the carousel via the API – that can only be done using the dashboard. If you are only using the first 3 spots then POST 'reset default' to IPaddress/api/config/splashbackground/4, the 5th location in the carousel will be populated with the default image and added to your carousel.

Calendar API

Solstice Host URLs: IPAddress/api/calendar *and* IPAddress/api/calendar/set *and* IPAddress/api/calendar/clear *and* IPAddress/api/calendar/add *and* IPAddress/api/calendar/delete

The calendar API allows an admin to send scheduling information to a Solstice host in a fully customizable way, without tying the host to a specific calendar via an Exchange server. To communicate with a host purely via the API, the calendar must be enabled and the calendar type set to '3rd-party only'.



There are two URLs that affect host endpoints when '3rd-party only' is selected as Calendar Type:

IPAddress/api/calendar/clear requires no data. Hitting this URL clears all calendar data.

IPAddress/api/calendar/set lets you POST JSON data about upcoming meetings. Each POST will override any existing calendar data, so the POST should be an array of all meeting/availability information that should show on the display.

The OpenControl API may also interact with a Microsoft Exchange calendar to integrate with an existing scheduling system. Once an Exchange account is authenticated through the Dashboard (Calendar Type = Microsoft Exchange), there are two URLs that allow you to add or remove a meeting on the Exchange calendar:

IPAddress/api/calendar/add requires the keys 'startTime', 'endTime', 'title', and 'organizer' to add a meeting to the Exchange calendar. A random 'id' key will be assigned when the meeting is created and may be used to delete the meeting.

IPAddress/api/calendar/delete requires only the 'id' key of the meeting to be deleted from the Exchange calendar.

To read calendar data from the display regardless of calendar type, GET the URL **IPAddress/api/calendar.**

Calendar Commands

calendarItems

Key	Type	Get/Post	Description
id	string	Get/Post	Unique meeting ID. Used internally only, does not show up on display. Must be supplied in '3rd-party' mode; automatically generated in 'Microsoft Exchange' mode.
startTime	long int	Get/Post	Meeting start time in Unix epoch seconds.
endTime	long int	Get/Post	Meeting end time in Unix epoch seconds.
title	string	Get/Post	Title of meeting that will show up on the display if meeting names are enabled in the Dashboard.
organizer	string	Get/Post	Name of meeting organizer that will show up on the display if enabled in the Dashboard.

Example POST data:

```
{
 "calendarItems": [
   {
     "id": "001",
     "startTime": 1767024030,
     "endTime": 1767031230,
     "title": "Engineering Review",
     "organizer": "Molly McNale"
   },
   {
     "id": "002",
     "startTime": 1767033000,
     "endTime": 1767038400,
     "title": "Plastics Lunch & Learn",
     "organizer": "Greg Clyff"
   }
 ]
```

Version and Update Control API

Solstice Host URLs:

- **IPAddress/api/version/currentversion** // returns JSON containing the field 'currentVersion' as a string.
- **IPAddress/api/version/updateavailable** // returns JSON containing the bool field 'isUpdateAvailable'. If this value is true, the string field 'updateAvailableTo' will be returned as well.
- **IPAddress/api/version/update** // updates Pod to version returned as 'updateAvailableTo' value.

Back to Top

Stats API

Solstice Host URL: IPAddress/api/stats

The stats API reports statistics about the current status of the Solstice host. These stats are instantaneous and can provide third-party developers with a snapshot of activity.

Global Stats Record

(Top Level)

Key	Type	Get/Post	Description
m_displayId	string	Get	This is the unique identifier that Solstice uses to manage a single instance, regardless of how the display is named or its current IP address
m_serverVersion	string	Get	The current software version running on the Solstice host.
m_displayInformation			
m_displayName	string	Get	The name of the display, shown on welcome screen and used for discovery.
m_productName	string	Get	The name of the Solstice software (Solstice).
m_productVariant	string	Get	The generation and type of Pod hardware as a name (Pod-only). For example, 'Gen1' or 'Pod Gen2'.
m_productHardwareVersion	int	Get	The Pod hardware generation as a version number (Pod-only). For example, 1 or 2.
m_statistics			
m_currentPostCount	int	Get	Total number of posts currently shared to the display.
m_currentBandwidth	int	Get	Total network bandwidth being used in Mbps.
m_connectedUsers	int	Get	Number of currently connected users.
m_timeSinceLastConnectionInitialize	int	Get	Time since the device last has a session initiated. Returns in milliseconds.
m_currentLiveSourceCount	int	Get	Number of current live sources such as USB camera.

Back to Top

Command API

Solstice Host URL: IPAddress/api/control

The Command API addresses runtime control of a Solstice host from a third-party application. Commands are executed by issuing a GET to the URL that corresponds to the command to be executed. The Command API does not make use of the JSON key/value records as the other APIs do.

Security is enforced by requiring password authentication when an administrator password has been set. If an incorrect or no password is appended to the GET when one is needed, the command will be ignored.
The URLs for each of the commands and their effects are listed below:

Command URL List

URL	Impact
/api/control/clear	Clears the display of all posts.
/api/control/boot	Boots all connected users and deletes all posts on the display. Returns display to splash-screen.
/api/control/reboot	Reboots host as soon as command is sent.
/api/control/restart	Restarts the Solstice software without reboothing the hardware.
/api/control/resetkey	Replace the current screen key with a new random screen key for connection authentication.

SDS API

SDS Server URL: IPAddress/api/discover

The SDS API reports statistics about the current status of the Solstice Discovery Service directory. These stats are instantaneous and can provide third-party developers with a snapshot of all hosts managed by this SDS server. Note that the target IP address must be the computer with SDS installed, not a Pod or Windows Host. While SDS may be installed on the same machine as a Windows Host, the API should not be used to call the server while the Solstice Software is active. In the Solstice Dashboard, this the IP address shown in the SDS tab:

Hitting a target URL of a Solstice host returns a single dictionary of key:value pairs pertaining to that one host. [sds]/api/discovery returns an array of dictionaries, where each dictionary contains the same keys but different values, depending on the values associated with the host in question. For example, sending a 'GET' to an SDS server with two associated hosts returns something similar to this:


```
{
    "displays": [
        {
            "name": "Pikes Peak PC",
            "id": "6cebfe86-3998-11e7-ab88-e09467b10fb1",
            "ipv4": "192.168.3.31",
            "port": 53100,
            "user_count": 0,
            "locked": false,
            "airplay_enabled": false,
            "session_capable": false,
            "in_session": false,
            "tags": []
        },
        {
            "name": "Solstice Demo 01",
            "id": "b242b840-4824-49d7-b5da-adb3a434a846",
            "ipv4": "192.168.3.31",
            "port": 53400,
            "user_count": 0,
            "locked": false,
            "airplay_enabled": false,
            "session_capable": false,
            "in_session": false,
            "tags": []
        }
    ]
}
```

SDS Commands

display

Key	Type	Get/Post	Description
name	string	Get	Assigned name of the Solstice host.
id	string	Get	This is the unique identifier that Solstice uses to manage a single instance, regardless of how the display is named or its current IP address.
ipv4	string	Get	IP address of Solstice host.
port	int	Get	Base port for host. The base port is always used along with the next two sequential ports.
user_count	int	Get	Number of users currently connected to the host.
locked	bool	Get	Checks if the display is locked
airplay_enabled	bool	Get	Checks if AirPlay mirroring is enabled for the display.
session_capable	bool	Get	Checks if the display is capable of starting a session. Would return 'false' if the host is a Windows PC that does not have the Solstice Software actively running.
in_session	bool	Get	Checks if host is involved in an active session, defined by one or more connections.
tags	array	Get	(Only shows if display has assigned tags). Returns array of dictionaries including the assigned tags' tag names ("tag_name":string) and color ("tag_color_index":int). There are four colors of tag: • 0 = blue • 1 = orange • 2 = green • 3 = pink
session_name	string	Get	Only shows if in_session=true. Returns nothing unless a session name has been assigned, such as for a Multi-Room meeting.
synced_display_ids	array	Get	Only shows if in_session=true. Returns nothing unless multiple displays are synced to the session, and then returns array of display IDs.

Back to Top

Valid Time Zones

The currently set time zone on a device is returned from the API call “timeZone” by id. The API call “timeZones” returns a string array of all available time zones.

id	name	offset (ms)	offset (hrs)
Pacific/Midway	GMT-11:00, Midway Island	-39600000	-11
Pacific/Honolulu	GMT-10:00, Hawaii	-36000000	-10
America/Anchorage	GMT-8:00, Alaska	-28800000	-8
America/Los_Angeles	GMT-7:00, Pacific Time	-25200000	-7
America/Tijuana	GMT-7:00, Tijuana	-25200000	-7
America/Phoenix	GMT-7:00, Arizona	-25200000	-7
America/Chihuahua	GMT-6:00, Chihuahua	-21600000	-6
America/Denver	GMT-6:00, Mountain Time	-21600000	-6
America/Costa_Rica	GMT-6:00, Central America	-21600000	-6
America/Regina	GMT-6:00, Saskatchewan	-21600000	-6
America/Chicago	GMT-5:00, Central Time	-18000000	-5
America/Mexico_City	GMT-5:00, Mexico City	-18000000	-5
America/Bogota	GMT-5:00, Bogota	-18000000	-5
America/Caracas	GMT-4:30, Venezuela	-16200000	-4.5
America/New_York	GMT-4:00, Eastern Time	-14400000	-4
America/Barbados	GMT-4:00, Atlantic Time (Barbados)	-14400000	-4
America/Manaus	GMT-4:00, Manaus	-14400000	-4
America/Halifax	GMT-3:00, Atlantic Time (Canada)	-10800000	-3
America/Santiago	GMT-3:00, Santiago	-10800000	-3
America/Sao_Paulo	GMT-3:00, Brasilia	-10800000	-3
America/Argentina/Buenos_Aires	GMT-3:00, Buenos Aires	-10800000	-3
America/Montevideo	GMT-3:00, Montevideo	-10800000	-3
America/St_Johns	GMT-2:30, Newfoundland	-9000000	-2.5
America/Godthab	GMT-2:00, Greenland	-7200000	-2
Atlantic/South_Georgia	GMT-2:00, Mid-Atlantic	-7200000	-2
Atlantic/Cape_Verde	GMT-1:00, Cape Verde Islands	-3600000	-1
Atlantic/Azores	GMT+0:00, Azores	0	0
Africa/Casablanca	GMT+0:00, Casablanca	0	0
Europe/London	GMT+1:00, London, Dublin	3600000	1
Africa/Windhoek	GMT+1:00, Windhoek	3600000	1
Africa/Brazzaville	GMT+1:00, W. Africa Time	3600000	1
Europe/Amsterdam	GMT+2:00, Amsterdam, Berlin	7200000	2
Europe/Belgrade	GMT+2:00, Belgrade	7200000	2
Europe/Brussels	GMT+2:00, Brussels	7200000	2
Europe/Sarajevo	GMT+2:00, Sarajev	7200000	2
Africa/Cairo	GMT+2:00, Cairo	7200000	2
Africa/Harare	GMT+2:00, Harare	7200000	2
Asia/Amman	GMT+3:00, Amman, Jordan	10800000	3
Europe/Athens	GMT+3:00, Athens, Istanbul	10800000	3
Asia/Beirut	GMT+3:00, Beirut, Lebanon	10800000	3
Europe/Helsinki	GMT+3:00, Helsinki	10800000	3
Asia/Jerusalem	GMT+3:00, Jerusalem	10800000	3
Europe/Minsk	GMT+3:00, Minsk	10800000	3
Asia/Baghdad	GMT+3:00, Baghdad	10800000	3

Europe/Moscow	GMT+3:00, Moscow	10800000	3
Asia/Kuwait	GMT+3:00, Kuwait	10800000	3
Africa/Nairobi	GMT+3:00, Nairobi	10800000	3
Asia/Tbilisi	GMT+4:00, Tbilisi	14400000	4
Asia/Yerevan	GMT+4:00, Yerevan	14400000	4
Asia/Duba	GMT+4:00, Dubai	14400000	4
Asia/Tehran	GMT+4:30, Tehran	16200000	4.5
Asia/Kabul	GMT+4:30, Kabul	16200000	4.5
Asia/Baku	GMT+5:00, Baku	18000000	5
Asia/Karachi	GMT+5:00, Islamabad, Karachi	18000000	5
Asia/Oral	GMT+5:00, Ural\'sk	18000000	5
Asia/Yekaterinburg	GMT+5:00, Yekaterinburg	18000000	5
Asia/Calcutta	GMT+5:30, Kolkata	19800000	5.5
Asia/Colombo	GMT+5:30, Sri Lanka	19800000	5.5
Asia/Katmandu	GMT+5:45, Kathmandu	20700000	5.75
Asia/Almaty	GMT+6:00, Astana	21600000	6
Asia/Rangoon	GMT+6:30, Yangon	23400000	6.5
Asia/Krasnoyarsk	GMT+7:00, Krasnoyarsk	25200000	7
Asia/Bangkok	GMT+7:00, Bangkok	25200000	7
Asia/Shanghai	GMT+8:00, Beijing	28800000	8
Asia/Hong_Kong	GMT+8:00, Hong Kong	28800000	8
Asia/Irkutsk	GMT+8:00, Irkutsk	28800000	8
Asia/Kuala_Lumpur	GMT+8:00, Kuala Lumpur	28800000	8
Australia/Perth	GMT+8:00, Perth	28800000	8
Asia/Taipei	GMT+8:00, Taipei	28800000	8
Asia/Seoul	GMT+9:00, Seoul	32400000	9
Asia/Tokyo	GMT+9:00, Tokyo, Osaka	32400000	9
Asia/Yakutsk	GMT+9:00, Yakutsk	32400000	9
Australia/Adelaide	GMT+9:30, Adelaide	34200000	9.5
Australia/Darwin	GMT+9:30, Darwin	34200000	9.5
Australia/Brisbane	GMT+10:00, Brisbane	36000000	10
Australia/Hobart	GMT+10:00, Hobart	36000000	10
Australia/Sydney	GMT+10:00, Sydney, Canberra	36000000	10
Asia/Vladivostok	GMT+10:00, Vladivostok	36000000	10
Pacific/Guam	GMT+10:00, Guam	36000000	10
Asia/Magadan	GMT+10:00, Magadan	36000000	10
Pacific/Majuro	GMT+12:00, Marshall Islands	43200000	12
Pacific/Auckland	GMT+12:00, Auckland	43200000	12
Pacific/Fiji	GMT+12:00, Fiji	43200000	12
Pacific/Tongatapu	GMT+13:00, Tonga	46800000	13

Back to Top