Overview

To better support the various business requirements of our customers our Universal Consent Platform (UCP) scripts have a supported API built in. This document outlines the functionality present in that API, how to use it, and how to solve some common scenarios.

What are the Universal Consent Scripts?

The UCP scripts power all the UCP functionality we provide on your web pages. They start with the script block you can retrieve from our Privacy Application. When that script block is added to your site it will start to function.

This script block is just a loader script, pulling in the actual scripts which power our consent functionality. The following scripts are always loaded:

evidon-site-notice-tag.js: this is the script which does the bulk of the work.

country.js: this script has no api functionality. It only allows us to detect the current country the user is in so we show the correct notice experience.

settings.js: this script contains the settings for the given domain and has no api functionality.

snthemes.js: this script contains the defined themes for your notices, which provide the look and feel of the displayed notice elements. It has no API functionality.

<language code>.js: this script provides all the messaging displayed to the user. The exact name of the file depends on the user's browser language settings. This has no API functionality.

In addition to the scripts listed above we have several scripts which can be loaded depending on other settings. They are:

evidon-banner.js: this script is downloaded if we need to display a consent banner to the current user. There are some supported API functions inside this script.

evidon-barrier.js: this script is downloaded if we need to display a consent barrier to the current user. There are some supported API functions inside this script.

General API Usage

When our scripts load they create a global object under the window namespace called “evidon”.
window.evidon

After our scripts successfully execute there will be a “notice” object added to the
window.evidon namespace.
window.evidon.notice

That is the starting point for our supported API. All API methods will be located on the window.evidon.notice object, or a sub-object (window.evidon.notice.banner for example).

If you inspect that object in something like Chrome tools you will see a bunch of methods located on the window.evidon.notice object. Some of them will start with an underscore (ex: _hookConsentEvents). Any method starting with an underscore is NOT a supported API method. Those indicate functionality we consider private or internal and should not be used by our customers.

Supported API Functionality

window.evidon.notice namespace

Properties

The following properties can be used from the window.evidon.notice object. These properties are provided to provide access to information that might be valuable to other scripts running on the page, but should be considered READ-ONLY. Any properties that can be modified at runtime will have a supported API method to allow that. For example, you can use the setLocation() method to change the country on the notice.

Property	Description
country	The currently active country object.
languageCode	The active language code.
languageRoot	The active root languge. This is used to pull the correct set of translation values.
activateTranslations	The object holding the currently active set of translations.
settings	The settings object holding all the possible settings for the current domain

domain	The active domain
activeSettings	The object holding the currently active settings for the notice.
consentTypeId	The identifier indicating the type of consent is active for this notice: 1: no consent 2: banner consent 3: barrier consent
privacyAccessTypeId	The identifier indicating the type of privacy tool access mechanism to show: 1: link 2: button

Methods

The following list of methods are supported under the window.evidon.notice namespace.

appendScript(url, callback)

url: the url to add to the script "src" attribute.
callback: an optional function to call once the script has finished loading.

Call this API method to append a script to the current document.

dropPixel(actionTypeId)

actionTypeId: the action identifier for the reporting ping to execute.

This method drops a reporting "ping" to the current document. This information is used to provide our customers insight into what the scripts are doing on their pages. In general we recommend against executing any reporting calls through your scripts, but if there is a legitimate need you can use this method to do it.

A "gotcha". We will only execute any reporting action one time. So if we have already recorded a specific action and you call this method again, it will not record the same action a second time.

consentGiven()

Calling this method will instruct our script to drop the consent cookie and execute any consent callback/tag manager functionality. Use this call if you are going to provide a consent mechanism not currently supported by our scripts (your own consent button for example).

getConsentUrl()

Returns the url to the consent tool for this notice. You can use this method to get the raw url, but we suggest using the "showConsentTool" method to actually display the consent tool.

showConsentTool()

Instructs our consent tool to be displayed based on the currently defined rules (overlay vs. new window for example). This is the recommended method to use if you are going to build your own access to our consent tool.

isMobile()

Just a helper method which returns true if we are running on a mobile device and false for a desktop. If you need to match any mobile vs. desktop logic from your scripts to ours we recommend using this method to check for mobile device execution.

showNotice()

Call this to display the notice. This is the primary method called to run all the logic and make all the checks necessary to display our notice elements based on the current environment settings (country, language, etc.). It would be unusual for any customer to need to call this from their script methods, but it is available if you need to.

setLocation(locationObj)

locationObj: the json object containing the details for the location you want to use. The format of the object is:

```
{'code':'us','id':1,'defaultLangauge':'en-us'}
```

This method is called to set the notice location (country). If you need to override the country settings for some reason you can use this method to accomplish that.

The following table contains the identifiers for all the countries we currently support.

1	us	United States	65	jo	Jordan
3	de	Germany	66	ir	Iran
4	es	Spain	67	ua	Ukraine
5	fr	France	68	vn	Viet Nam
6	gb	United Kingdom	69	ws	Samoa

7	it	Italy	70	am	Armenia
8	nl	Netherlands	71	ph	Philippines
9	ca	Canada	72	al	Albania
10	cn	China	73	ai	Anguilla
11	il	Israel	74	ag	Antigua & Barbuda
12	mx	Mexico	75	aw	Aruba
13	ru	Russia	76	bs	Bahamas
15	dk	Denmark	77	bb	Barbados
16	ie	Ireland	78	bz	Belize
17	no	Norway	79	bm	Bermuda
18	pl	Poland	80	bo	Bolivia

19	at	Austria	81	ba	Bosnia & Herzegovina
20	be	Belgium	82	vg	British Virgin Islands
21	ch	Switzerland	83	ky	Cayman Islands
22	se	Sweden	84	cl	Chile
23	sk	Slovakia	85	co	Colombia
24	hu	Hungary	86	cg	Congo (Brazzavilla)
25	fi	Finland	87	cr	Costa Rica
26	pt	Portugal	88	cd	Democratic Republic of the Congo
27	cz	Czech Republic	89	dm	Dominica
28	lu	Luxembourg	90	do	Dominican Republic
29	gr	Greece	91	ec	Ecuador
30	bg	Bulgaria	92	sv	El Salvador
31	ro	Romania	93	ga	Gabon
32	ee	Estonia	94	gh	Ghana
33	lv	Latvia	95	gd	Grenada
34	lt	Lithuania	96	gt	Guatemala
35	si	Slovenia	97	gy	Guyana
36	mt	Malta	98	ht	Haiti
37	cy	Cyprus	99	hn	Honduras
38	is	Iceland	100	jm	Jamaica
39	tr	Turkey	101	md	Moldova
40	eg	Egypt	102	ma	Morroco
41	sa	Saudi Arabia	103	an	Netherlands Antilles
42	br	Brazil	104	ni	Nicaragua
43	ar	Argentina	105	pa	Panama
44	tw	Taiwan	106	py	Paraguay

45	kr	South Korea	107	pe	Peru
46	jp	Japan	108	pr	Puerto Rico
47	au	Australia	109	kn	Saint Kitts & Nevis
48	nz	New Zealand	110	sn	Senegal
49	za	South Africa	111	lc	Saint Lucia
50	dz	Algeria	112	vc	Saint Vincent & Grenadines
51	ly	Libya	113	sr	Suriname
52	sd	Sudan	114	tt	Trinidad & Tobago
53	ng	Nigeria	115	tn	Tunisia
54	ae	United Arab Emirates	116	tc	Turks & Caicos Islands
55	id	Indonesia	117	uy	Uruguay
56	my	Malaysia	118	vi	Virgin Islands
57	th	Thailand	119	ve	Venezuela
58	pk	Pakistan	120	ge	Georgia
59	in	India	121	az	Azerbaijan
60	sg	Singapore	122	la	Laos
61	hr	Croatia	123	kh	Cambodia
62	mk	Macedonia	124	np	Nepal
63	rs	Serbia	125	lk	Sri Lanka
64	hk	Hong Kong	126	mm	Myanmar
			127	bn	Brunei

setThemes(themes)

themes: object structure containing the theme details you want to use.

It is highly recommended you not use this method. The theme structure is relatively complex and is identifier based, so you would need to match everything up perfectly. However, if you need to tackle this for some reason it is available.

activateTranslations(code)

code: language code to make the active translation set

Use this method if you want to force a specific language set that is different from the user browser settings. A typical example would be a site that uses a dropdown to switch between languages where you want to force the consent messaging to match your site instead of the browser settings.

loadSettings(settingsObj)

settingsObj: the json object structure containing the settings to use for the current notice.
This is another method that is available, but we highly recommend not using. The setting structure is complex and needs to correlate the identifiers between the various translations and theme elements.

getBannerStyle()

Use this method if you need to pull the styles for the banner object.

getBarrierStyle()

Use this method if you need to pull the styles for the barrier object.

getButtonStyle()

Use this method if you need to pull the styles for the consent tool button object.

getLinkStyle()

Use this method if you need to pull the styles for the consent tool link object.

getTranslations()

Returns the active set of translations.

window.evidon.banner namespace

The following list of methods are supported under the window.evidon.banner namespace.

IEVersion()

Helper method called to return the current version of IE the scripts are running on. We have some conditional code blocks which execute differently on earlier version of IE then more modern browsers.

closeBanner(eventObj)

eventObj: a standard event object received from javascript click events.

This is the click handler from the banner accept button and close icon. You can call it from your own code if you want to simulate a click event but it is recommended you use the window.evidon.notice.consentGiven() api instead.

acceptButtonClicked(eventObj)

eventObj: a standard event object received from javascript click events.

This is the callback executed when someone clicks our Accept button. If you need to simulate an Accept button click, use this method.

optionButtonClicked(eventObj)

eventObj: a standard event object received from javascript click events.

This is the callback executed when a user clicks our Options button on the banner. If you need to simulate an Option button click call this method.

closeIconClicked(eventObj)

eventObj: a standard event object received from javascript click events.

The callback executed when someone click the close icon on the banner. If you need to simulate the close icon click call this method.

endsWith(source,value)

source: source string to check
value: value to look at the end of the source string for

This is just a helper method which returns a Boolean true if the source string ends with the passed in value.

isOnBanner(element)

element: An element object

This method is a helper method that returns true if the given element object is actually located on our banner display. It is used to keep us from executing a consent event if someone clicks at some random place on the banner. It is unlikely you will need to use this method for anything.

closeConsentBanner()

Instructs the banner to close. This method does not execute any consent functionality. It should only be used if you want to hide the banner without a user consenting for some reason.

getBannerStyle()

Returns the style used to format the banner on the page. This is just a helper method which makes sure some specific values (z-index for example) are present in the styles being applied to the banner.

createAcceptButton()

Just a helper method which creates a formatted Accept button. There probably isn't a good use case for a customer using this, but it's there if you need it.

createOptionButton()

A helper method to create the option button displayed on the banner.

createCloseIcon()

A helper method to create the close icon displayed on the banner.

createBanner()

Method called when the script loads to create the banner structure.

setTextValues(translations)

translations: an object containing the string values for the various banner elements.

This is called to apply the given translations to the banner. Typically this is called from our main notice script if the language is changed. It is highly recommended you just use our user interface to make any translation changes, but if you need to force these to be different for some reason you can use this method.

createNotice()

Called to create the notice and show it on the page. This is called as soon as the script loads, so it is unlikely you will need to use it, but if you have to force the banner to display this method will do that.

destroyNotice()

Instructs the notice to close and remove itself from the dom. Calling this method does not trigger any type of consent actions or drop the consent cookie. You can use this method if you need to remove the banner from the current page for some reason.

scaleForMobile()

This is a helper method called to apply a zoom on the banner if displayed on a mobile device. It is unlikely this method needs to be used by any customers.