VELVENTI API

USER GUIDE

CADS

AUTODESK. Authorised Developer

Microsoft Partner Gold Application Development

The VelVenti API is a cloud-based RESTful web service that returns wind load sector data, such as the peak velocity pressure, the orography factor and the distance to sea, in an XML or JSON format for a specified site located anywhere in the world. At present, the API also returns the ground and characteristic snow load data for UK and Irish sites.

This document is intended for IT staff, particularly software engineers, to help them understand how to integrate the VelVenti API into existing applications, such as Microsoft Excel-based reporting tools, software programs running on a computer, or web applications. This user guide describes how to call the VelVenti APIs, and the various parameters involved.

API Call Formulation

A VelVenti API call is structured as follows:

```
URL/ApiName?apiKey=unique-key&param1=value1&param2=value2&...&paramN=valueN
```

Where `value`*i*, *i*=1,…,*N* are values for the various input parameters, where applicable.
The URL is the designated web address. For example:

http://www.velventi.com/api/v1.1/fetch/BasicReport?apiKey=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX&rf=xml&lat=50.763&lng=-1.995&bh=28

In this call, the various parts are:

URL:	http://www.velventicom/api/v1.1/fetch
API Name:	**BasicReport**
API Key:	**XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX**
Parameters:	**& rf=xml&lat=50.763&lng=-1.995&bh=28**

An example of an API Licence key is: **CE606B90-E1ED-43BF-8156-E0B8AF7F76A9**. Note that this key is no longer operational.

The above call instructs the VelVenti API to return a basic report for a structure whose height is 28 metres and which is located at latitude=50.763 and longitude=-1.995. The latter values may also be specified with a combined parameter: &latlng=50.763|-1.995. Note the vertical bar character that separates the two values. The result of the call is returned in XML format (rf=xml), as shown below, but the more-compact JSON format may also be obtained by specifying (&rf=json). The textual output, which is easily readable, is typically processed by a computer program or used in an Excel macro. Each result contains a service status, that includes the version number of the VelVenti API service and an error message.

```
<BasicReport>
      <!--
      BasicReport generated by VelVenti API on Wednesday, January 24, 2018 at
      3:02:37 PM.
      -->

      <ServiceStatus>
            <Name>CADS VWS</Name>
            <Version>2.0.97</Version>
            <Error/>
            <Query>&rf=xml&lat=50.763&lng=-1.995&bh=28</Query>
            <Reference>AA02FB5A-D8EA-44E1-9F12-D682726CDC2F</Reference>
      </ServiceStatus>
      <SiteInformation>
            <Latitude>50.763</Latitude>
            <Longitude>-1.995</Longitude>
            <BuildingHeight>...</BuildingHeight>
            <BuildingOrientation>...</BuildingOrientation>
            <SiteAltitude>...</SiteAltitude>
            <BasicWindSpeed>...</BasicWindSpeed>
            <BasicSnowLoad>...</BasicSnowLoad>
            <SeasonalFactor>...</SeasonalFactor>
            <ProbabilityFactor>...</ProbabilityFactor>
            <DesignCode>BsEn199114</DesignCode>
            <WindLoadIntended>Whole</WindLoadIntended>
            <DesignLife>UpTo50Years</DesignLife>
      </SiteInformation>
      <WindSectorsDetails>
            <Description>12 sectors of 30°</Description>
            <Count>12</Count>
            <SectorAngle>30</SectorAngle>
            <WindDirections>
                  <double>0</double>
                  <double>30</double>
                  <double>60</double>
                  <double>90</double>
                  <double>120</double>
                  <double>150</double>
                  <double>180</double>
                  <double>210</double>
                  <double>240</double>
                  <double>270</double>
                  <double>300</double>
                  <double>330</double>
            </WindDirections>
            <DominantIndex>9</DominantIndex>
            <DominantDirection>270</DominantDirection>
            <AngleUnit>°</AngleUnit>
      </WindSectorsDetails>
      <DirectionalFactor>...</DirectionalFactor>
      <AltitudeFactor>...</AltitudeFactor>
      <LocalEffectsFactor>...</LocalEffectsFactor>
      <BasicWindVelocity>...</BasicWindVelocity>
      <BasicVelocityPressure>...</BasicVelocityPressure>
      <OrographyCategory>...</OrographyCategory>
      <IsOrographySignifican>false</IsOrographySignifican>
      <OrographyFactor>...</OrographyFactor>
      <TerrainCategory>...</TerrainCategory>
      <ExposureFactor>...</ExposureFactor>
      <ExposureCorrectionFactor>...</ExposureCorrectionFactor>
      <DistanceToSea>...</DistanceToSea>
      <DistanceInsideTown>...</DistanceInsideTown>
      <ZoneForSizeFactor>...</ZoneForSizeFactor>
      <PeakWindVelocity>...</PeakWindVelocity>
      <PeakVelocityPressure>
            <Name>Peak Velocity Pressure</Name>
            <Value>0.82323938262252494</Value>
            <Unit>kN/m²</Unit>
            <Symbol>q<sub>p</sub></Symbol>
            <Reference>NA.2.17</Reference>
            <IsUserSpecified>False</IsUserSpecified>
      </PeakVelocityPressure>
</BasicReport>
```

As can be seen above, each returned parameter, e.g. Peak Velocity Pressure, might have a unit, a symbol, a reference in the corresponding wind load standard document, and an indication about whether the parameter value was specified by the user.

List of input parameters.

Apart from the location of the building structure, all other input parameters are optional. However, where unspecified, the following default values are used:

- Building height is 20 m
- Building orientation from north: 0 degrees
- Design life is 50 years
- Wind load design code is BS EN 1991-1-1 for UK sites

The following table lists the complete set of codes for identifying input parameters. The naming convention is straightforward. The majority of parameter codes are made up of the first letter of the key words used to refer to the parameter: For example, the codes for `building height and` (wind load`)` `design code` are `bh` and `dc`, respectively

Code	Input Parameter	Type	Default Value	Note
rf	API Return Format	String	XML	XML | JSON
lat	Latitude	Double		e.g. 50.763
lng	Longitude	Double		e.g. -1.995
latlng	Latitude & longitude	String	value|value	e.g. 50.762|-1.993
ogr	OS Grid Reference	String		If OS Grid Ref is specified, lat/lng are ignored.
pc	Postcode	String		For UK, must be compliant
dc	Wind Load Design Code	String	BsEn199114	BsEn199114 | IsEn199114 | Bs6399 | Asce710
cm	Calculation Method	String	Standard	For BS 6399-2: Standard | Directional
wli	Wind Load Intended	String	Whole	Whole | Individual
bh	Building Height	Double	20	In metres
bo	Building Orientation	Double	0	In degrees from north, increasing clockwise. Optional. Default is 0°
dl	Design Life	Integer	50	A number between 1 and 120 (in years)
wst	Wind Sector Type	Integer	12	1| 4| 8| 12, depending on the design code
tc	Terrain Category	String	Country	Country | Town | Sea | EdgeOfTown
atd	Apply Terrain Data	Boolean	true	Use the same terrain data for all wind sectors
citb	Consider Influence of Tallest Nearby Building	Boolean	false	If true, then the corresponding length, height, and radial distance need to be specified for all sectors or individual sectors
ltb	Length of Tallest Building	Double	50	In metres
htb	Height of Tallest Building	Double	20	In metres
rtb	Radial Distance from Tallest Building	Double	5	In metres
ise	Include Sheltering Effects of Surrounding Buildings	Boolean	false	If true, then the average height of surrounding buildings need to be specified

ahsb	Average Height of Surrounding Buildings	Double	8	In metres
adsb	Average Distance to Surrounding Buildings	Double	12	In metres
af	Altitude factor	double		
bws	Basic wind speed	double		In m/s
df	Directional factor	double		
sf	Seasonal factor	double		
sa	Site altitude	double		
pf	Probability factor	double		
oc	Orography category	string		
of	Orography factor	double		Flat | HillOrRidge | CliffOrEscarpment°
ef	Exposure factor	double		
ecf	Exposure correction factor	Double		
laf	Local effects factor	double		
dts	Distance to sea	double		
dit	Distance inside town	double		
sws	Selected wind sector	int		Typically between 1 and 12 (design code depending)
ih	Intermediate heights	Array of doubles	e.g. 5|10|13|18	To calculate the peak velocity pressure at different levels along the building height

If a grid reference or a postcode is specified as well as latitude and longitude values, then the latter are ignored in favour of the grid reference or postcode. The precedence is grid reference, postcode, and lat/lng values.

If a parameter is specified more than once, then the last parameter is used. All parameters are validated prior to processing.

If a parameter consists of an array of values, for example the height of the tallest structure in various wind sectors, then the values must be separated with a vertical bar character, e.g. 25.5 | 16.9 | | 20.3|22.7. Where a value is missing (as between the two vertical bars in this example), the API will treat it as unspecified by the user, and therefore the default value is used instead.

API Names

At present, the following names are available: **Version**, **BasicReport** and **FullReport**. The *BasicReport* API returns wind data for the dominant wind sector only, whereas the *FullReport* API returns wind data for all the wind sectors. The *Version* API does not need any additional parameters, apart from the API key. More dedicated APIs will be added as the need arises to turn the VelVenti APIs into a comprehensive web service for wind and snow load calculations.

List of output parameters

The list of returned parameters depends on the API used. The BasicReport returns less data (just about the dominant sector), whereas the FullReport returns more data, about all the wind sectors.

Code	Output Parameter	Example	Unit	Symbol	Reference
	Service Status				
	Name	CADS VWS			
	Version	e.g. 2.0.97			
	Error	Empty if the API call was successful.			
	Query	The actual list of input parameters, as a string.			
	Reference	Unique ID that references the API call			
	Site Information				
	Latitude	50.763			
	Longitude	-1.995			
	BasicWindSpeed	21.475	V_b	m/s	Figure NA.1
	BasicSnowLoad	0.50	pg	kN/m^2	Fig. 7.1
	BuildingHeight	28	H	m	
	BuildingOrientation	0		°N	
	SiteAltitude	31.890	A	m	
	SeasonalFactor	1.00	C_{season}		Table NA.2
	ProbabilityFactor	1.00	C_{prob}		NA.2.8
	WindLoadDesignCode	BsEn199114			
	WindLoadIntended	Whole			
	DesignLife	50			
	Wind Sector Information: All sectors with FullReport API. Dominant Sector with BasicReport API				
	WindSectorsDetails: `Count | SectorAngle | WindDirections | DominantIndex | DominantDirection | AngleUnit`				
	DirectionalFactor: `Name | Value | Unit | Symbol | Reference | IsUserSpecified`				
	AltitudeFactor				
	LocalEffectsFactor				
	BasicWindVelocity				
	BasicVelocityPressure				
	SiteWindSpeed				
	FetchFactor				
	FetchCorrectionFactor				
	GustWindSpeed				
	MeanWindVelocity				
	OrographyCategory				
	IsOrographySignificant				

	OrographyFactor
	OrographyCorrectionFactor
	TerrainCategory
	ExposureFactor
	ExposureCorrectionFactor
	DistanceToSea
	DistanceIntoTown
	AverageHeightOfSurroundingStructures
	AverageDistanceToSurroundingStructures
	DisplacementHeight
	EffectiveHeight
	OverallHeightOfTheHighRiseBuilding
	OverallLengthOfTheHighRiseBuilding
	RadialDistanceFromHighRiseBuilding
	RoughnessFactor
	RoughnessCorrectionFactor
	TurbulenceFactor
	TurbulenceCorrectionFactor
	ZoneForSizeFactor
	PeakWindVelocity
	PeakVelocityPressure

The above table lists all possible output parameters. However, the returned report will only contain applicable parameters.

Overriding Output Parameters

The value of an output parameter may be overridden by specifying it in the list of input parameters. For example, the site altitude may be specified if a more accurate value is known. To do so, simply add &sa=value (e.g. &sa=25.6). VelVenti API will then use that value in any of the related calculations it performs. When results are returned, the overridden parameter will be listed with the same value, but marked as specified by the user (see `IsUserSpecified=true`).

Manually Specifying Different Terrain Categories

To adjust the terrain category of a given wind sector, thus overriding VelVenti's estimate, use parameter code tc. Let's assume, that we have a 12-sector configuration, and the user wishes to specify *Country* along wind sectors 2 and 3 and Sea along wind sector 7, in the clockwise direction. The user would need to specify the following input parameter: **&tc=|Country|Country|||Sea**. Note that unspecified values will be replaced by VelVenti's estimates. Also, note that as not all the array values are specified, VelVenti will use its estimates for the remaining values.

Peak Velocity Pressure at Intermediate Building Heights

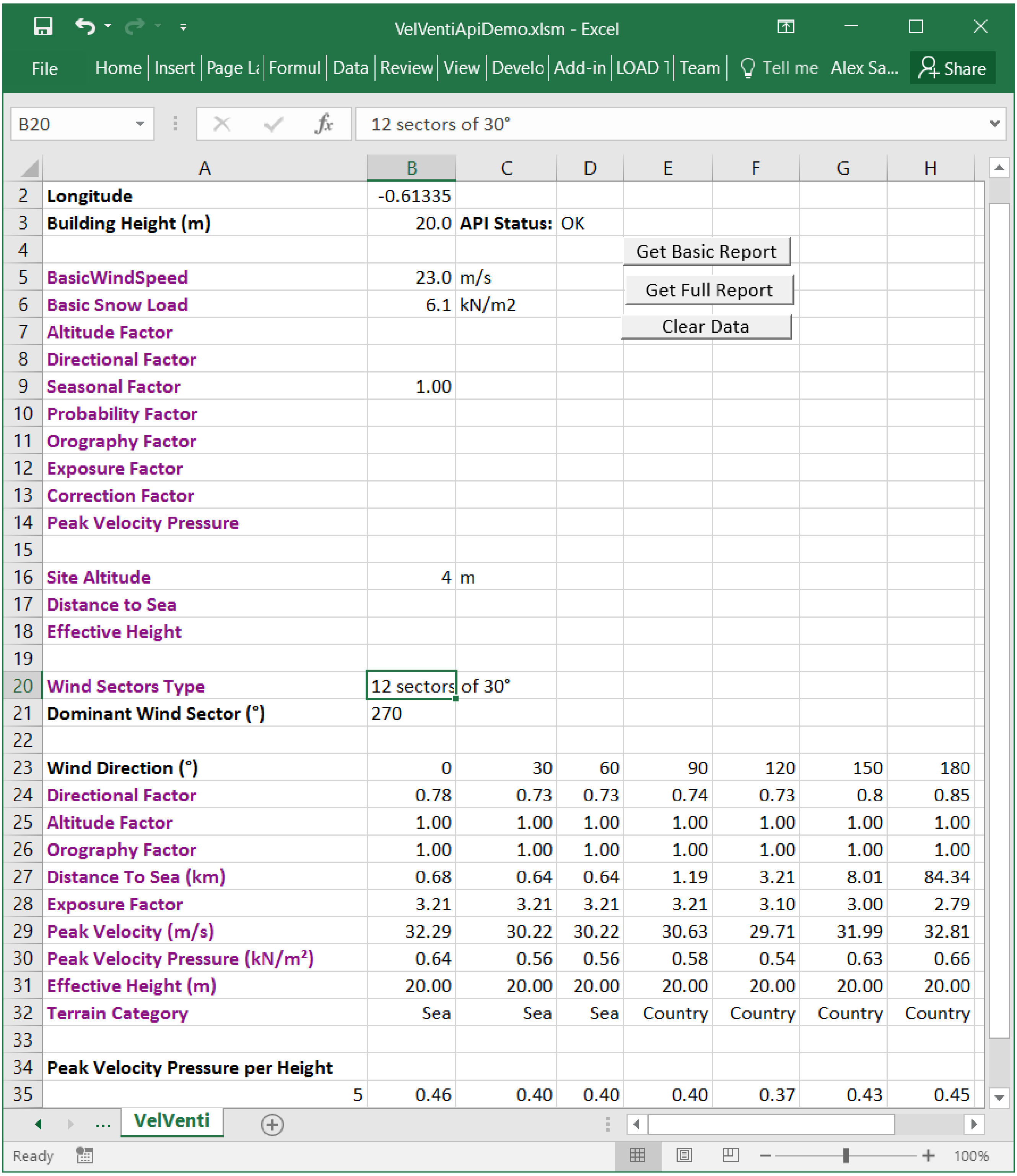
By default, VelVenti calculates the peak velocity pressure at predefined levels along the building height, typically at 5m, 10m, 20m, 30m, etc, all the way up to the actual building height. However, the user may specify the actual levels of interest by using parameter code **ih** for Intermediate Heights. For example &ih=3|5|8|12|16 for a building with a height of 20 m. Note that VelVenti will rearrange the values in ascending order and ignore any value greater than the building height.

MS-Excel Integration

The VelVenti API can be integrated into MS-Excel by means of a Visual Basic macro. The following code snippet illustrates the case for a basic report. Essential data, such as the latitude and longitude may be read from cells, such as B2 and C2, and the output could be processed or displayed in dedicated cells as part of a reporting tool, e.g. B4 and B12.

```
Sub BasicReport_Click()
    Dim ApiRequest As String
    ApiRequest = "http://www.velventi.com/api/v1.1/fetch/BasicReport?apiKey="
& Range("F1").Value _
               & "&rf=xml" _
               & "&lat=" & Range("B2").Value _
               & "&lng=" & Range("C2").Value _
               & "&bh=" & Range("D2").Value

    Dim MyRequest As Object
    Set MyRequest = CreateObject("WinHttp.WinHttpRequest.5.1")
    MyRequest.Open "GET", ApiRequest

    'Send API Request.
    MyRequest.send

    Dim xDoc As Object
    Set xDoc = CreateObject("MSXML2.DOMDocument")
    xDoc.async = False: xDoc.validateOnParse = False
    xDoc.LoadXML (MyRequest.responseText)

    Range("B4").Value = CDbl(
xDoc.SelectNodes("//SiteInformation/BasicWindSpeed/Value")(0).Text)
    Range("C4").Value =
xDoc.SelectNodes("//SiteInformation/BasicWindSpeed/Unit")(0)
    Range("B12").Value = CDbl(
xDoc.SelectNodes("//PeakVelocityPressure/Value")(0).Text)
    Range("C12").Value =
xDoc.SelectNodes("//PeakVelocityPressure/Unit")(0).Text
      ‘ Etc for other parameters

      ‘ In some cases, it may be necessary to test whether a
      ‘ parameter is present before attempting to access it
    Dim node As IXMLDOMNode
    Set node = xDoc.SelectSingleNode("//ExposureCorrectionFactor")
    If Not node Is Nothing Then
    Range("B13").Value = CDbl(
xDoc.SelectNodes("//ExposureCorrectionFactor/Value")(0).Text)
    End If
End Sub
```

The following image shows the data on the spreadsheet, where the API is triggered by means of a push button that calls the underlying Visual Basic macro.

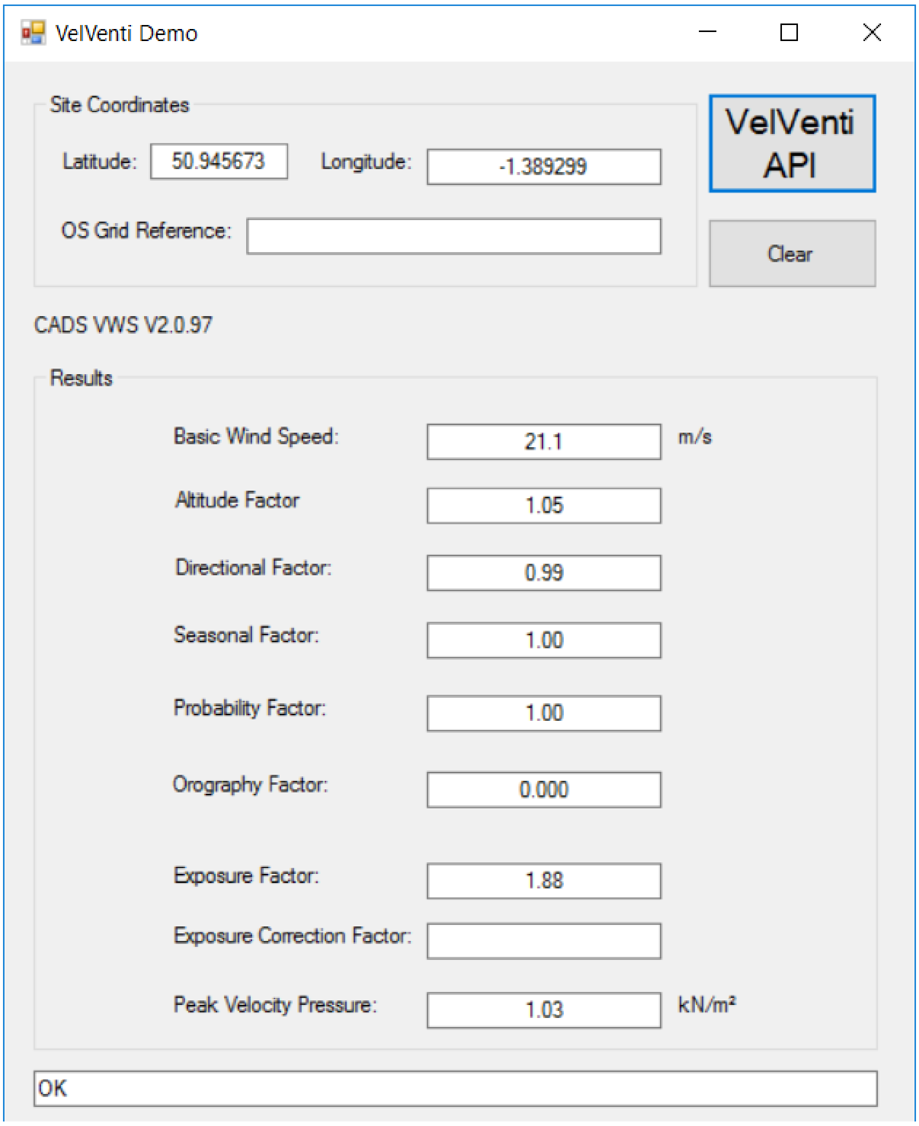


	A	B	C	D	E	F	G	H
2	**Longitude**	-0.61335						
3	**Building Height (m)**	20.0	**API Status:**	OK				
4								
5	BasicWindSpeed	23.0	m/s					
6	Basic Snow Load	6.1	kN/m2					
7	Altitude Factor							
8	Directional Factor							
9	Seasonal Factor	1.00						
10	Probability Factor							
11	Orography Factor							
12	Exposure Factor							
13	Correction Factor							
14	Peak Velocity Pressure							
15								
16	Site Altitude	4	m					
17	Distance to Sea							
18	Effective Height							
19								
20	Wind Sectors Type	12 sectors of 30°						
21	**Dominant Wind Sector (°)**	270						
22								
23	**Wind Direction (°)**	0	30	60	90	120	150	180
24	Directional Factor	0.78	0.73	0.73	0.74	0.73	0.8	0.85
25	Altitude Factor	1.00	1.00	1.00	1.00	1.00	1.00	1.00
26	Orography Factor	1.00	1.00	1.00	1.00	1.00	1.00	1.00
27	Distance To Sea (km)	0.68	0.64	0.64	1.19	3.21	8.01	84.34
28	Exposure Factor	3.21	3.21	3.21	3.21	3.10	3.00	2.79
29	Peak Velocity (m/s)	32.29	30.22	30.22	30.63	29.71	31.99	32.81
30	Peak Velocity Pressure (kN/m²)	0.64	0.56	0.56	0.58	0.54	0.63	0.66
31	Effective Height (m)	20.00	20.00	20.00	20.00	20.00	20.00	20.00
32	Terrain Category	Sea	Sea	Sea	Country	Country	Country	Country
33								
34	**Peak Velocity Pressure per Height**							
35	5	0.46	0.40	0.40	0.40	0.37	0.43	0.45

Desktop Application Integration

The following code snippet illustrates how the VelVenti API could be called from a C# code. Note that the try/catch syntax is omitted here for simplicity.

```
private async Task CallVelVentiAPI()
{
    var client = new HttpClient();

    client.BaseAddress = new Uri("http://www.velventi.com/");
    client.DefaultRequestHeaders.Accept.Clear();
    client.DefaultRequestHeaders.Accept.Add(new 
MediaTypeWithQualityHeaderValue("application/json"));

    var apiRequest = @"api/v1.1/fetch/BasicReport?apiKey=" + _apiKey + "&rf=json";
    if (string.IsNullOrWhiteSpace(textBoxOGR.Text.Trim()))
    {
      if (string.IsNullOrWhiteSpace(textBoxLat.Text.Trim()) || 
string.IsNullOrWhiteSpace(textBoxLng.Text.Trim())) return;

       apiRequest += @"&lat=" + textBoxLat.Text.Trim();
       apiRequest += @"&lng=" + textBoxLng.Text.Trim();
   }
   else
   {
       apiRequest += @"&ogr=" + textBoxOGR.Text.Trim();
   }

   HttpResponseMessage response = await client.GetAsync(apiRequest);
   if (response.IsSuccessStatusCode)
   {
       var jsonResponse = await response.Content.ReadAsStringAsync();

       var dynResponse = JObject.Parse(jsonResponse);
       var apiError = (string)dynResponse["ServiceStatus"]["Error"];
       if (!string.IsNullOrWhiteSpace(apiError))
       {
           textBoxError.Text = apiError;
           return;
       }
       textBoxError.Text = "OK";

        labelProgress.Text = (string)dynResponse["ServiceStatus"]["Name"] + " V" + 
(string)dynResponse["ServiceStatus"]["Version"];

        var doubleValue = 
(double)dynResponse["SiteInformation"]["BasicWindSpeed"]["Value"];
        textBoxBWS.Text = doubleValue.ToString("0.0");
        labelBWS.Text = 
(string)dynResponse["SiteInformation"]["BasicWindSpeed"]["Unit"];

        doubleValue = (double)dynResponse["AltitudeFactor"]["Value"];
        textBoxAF.Text = doubleValue.ToString("0.00");
        //etc.
    }
    else
    {
        textBoxError.Text = "The VelVenti API call failed";
    }
}
```



Site Coordinates

Latitude: 50.945673 Longitude: -1.389299

OS Grid Reference:

CADS VWS V2.0.97

Results

Basic Wind Speed:	21.1	m/s
Altitude Factor	1.05	
Directional Factor:	0.99	
Seasonal Factor:	1.00	
Probability Factor:	1.00	
Orography Factor:	0.000	
Exposure Factor:	1.88	
Exposure Correction Factor:		
Peak Velocity Pressure:	1.03	kN/m²

OK

Please contact CADS Support for any help with the above on +44 (0)1202 603733