GlobeSpotter

API specification

GlobeSpotter API version 2.8

Table of Contents

Version history

Version	Status	Date	Details of change	Author(s)
0.1	Draft	20100908	Initial version	SBr
0.5	Draft	20100923	Reworked. Added core API functionality	SBr
0.9.0	Draft	20101007	Applied corrections & improvements. Added to core API. Added Multi-	RBu
0.9.1	Pre-Release	20101013	Fixed some omissions.	RBu
0.9.2	Release	20101013	Added permission description	SBr
1.0	Release	20101015	Added flash player requirements	SBr
1.1	Release	20110328	Added MDI, Full API, measurement functions and events	SBr
1.1.1	Release	20110401	Added permissions table	SBr
1.2	Draft	20111221	Document 2.6 API: Compatibility list, simplified manual	SBr, JBr
1.3	Draft	20120102	Document 2.6 API	JBr
1.4	Draft	20120113	Added ViewerTitle(Bar)-settings + TilingMode + result on closeMeasurement()	ROo
1.5	Draft	20120224	Added map properties + functionality + settings	ROo
1.6	Release	20120401	Added Compatibility list. Corrected paragraph 1.3	Sbr
1.6	Release	20120420	Fixed URLs referring to the API, removed incorrect examples and referred to the CM webpage. Fixed the table of contents.	JBr
1.6.1	Release	20120515	Simple HTML example updated. Parameter SRSNameMap requirement updated.	JBr
1.6.2.	Release	20120605	Updated the setServiceURL function reference.	JBr
1.7	Draft	20121126	Added new 2.7 functions	ROo
1.7.1	Release	20121204	Added missing function descriptions	ROo
1.7.1.1	Draft	20130207	Removed: AddOSMLayer	HBr

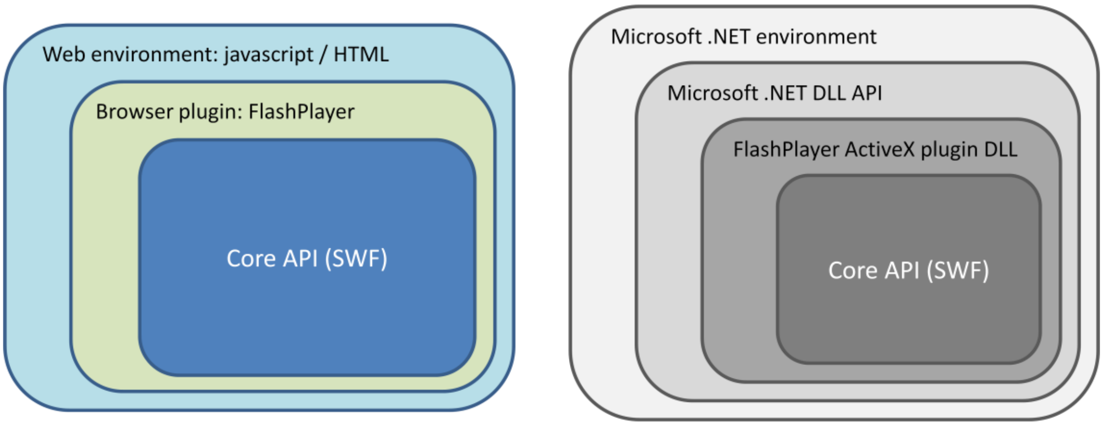
1.7.1.2	Draft	20130211	Removed: SetSRSNameMap, Renamed: GetSwapViewersEnabled to SetSetSwapViewersEnabled	HBr
1.7.1.3	Release	20130325	Added get-/setAddressDatabase, added getSupportedAdressDatabases	ROo
1.8	Release	20130619	Updated: getEntityData, setServiceURL, hst_measurementPointObservationAdded, hst_mapInitialized Added: autoComplete, get-/setMeasurementSmartClickModeEnabled, hst_autoCompleteResult, Added for oblique support: get-/setObliqueModeEnabled, get-/setMapObliqueDirectionsEnabled, get-/setMapObliqueToggleVisible, obliqueViewFromBehind, obliqueViewFromLeft, obliqueViewFromRight	ROo
1.8.1	Draft	20130916	Added: get-/setDateFormat, get-/setTimeFormat Promoted Oblique-functions to release version	ROo
1.8.2	Release	20130925	Added: hst_showLocationRequested	ROo
1.8.3	Release	20131118	Added: hst_detailImagesVisibilityChanged, get-/setUnitAreaFactor, get-/setUnitAreaLabel, get-/setUnitAreaPrecision, get-/setUnitLengthFactor, get-/setUnitLengthLabel, get-/setUnitLengthPrecision, get-/setUnitVolumeFactor, get-/setUnitVolumeLabel, get-/setUnitVolumePrecision	

1 Introduction

This document describes the GlobeSpotter API. It can be used by integrators to control one or multiple cyclorama windows embedded within a web environment or a desktop application.

1.1 Integration environments

Integration is possible within an AS3, web, and Microsoft .NET environment. The following image shows the components involved in a web and Microsoft .NET integration:


Web environment: javascript / HTML
Browser plugin: FlashPlayer
Core API (SWF)
Microsoft .NET environment
Microsoft .NET DLL API
FlashPlayer ActiveX plugin DLL
Core API (SWF)

The API is packed in a Flash SWF application. When used within a web environment, it is possible to use either HTML for simple applications or use HTML in combination with JavaScript to create more complex applications. When combining the API with the .NET API DLL, it is possible to create complex applications within the Microsoft .NET environment.

1.2 System requirements

Hardware	Minimum 1GHz with 256MB memory
Operating system	Windows, Macintosh, Linux, Solaris, Citrix
Browser	Internet Explorer, Mozilla Firefox, Opera, Apple Safari Note: Google chromium only works if user credentials are set during initialization of the API.
RIA Platform	All API packages use Adobe's Flash player. In order to use the GlobeSpotter API, Adobe Flash Player version 10.0.22.18 or higher is required. The player can be installed from the Adobe website. On the website you can find the installation instructions.
Connection	Internet access

During integration development it's highly recommended to use the debug player to identify problems caused by the web services, API or integration software.

For all web integrations, it's the user's choice which browser will be used and also which flash player will be used (Internet explorer, Mozilla/Netscape).

The Microsoft Windows .NET integration always uses the Internet explorer flash player as an ActiveX plug-in.

1.3 Revision structure

The GlobeSpotter API is available on the following location:

v2.8:

```
  https://globespotter.cyclomedia.com/v28/api/viewer_api.swf
```

It's highly recommended to download the SWF each time the integration is started. The advantage of downloading the SWF is always having the latest (minor) revision of the GlobeSpotter API. New major revisions of the GlobeSpotter API will have a different url (`/v29/api/..`).

1.4 API license and permissions on functionalities

Depending on the GlobeSpotter license of a user, he or she might be allowed limited access to various API functionalities. The `getPermissions` API function returns the permissions on API functionalities based on the user's credentials. These permissions can be any of the following:

Permission	*Pro license required*	*Description*
Basic	No	If the user does not have basic permissions, no use of the API is allowed.
AddLayerWMS	Yes	Allowed to add WMS layers.
AddLayerWFS	Yes	Allowed to add WFS and GML layers.
MeasureLine	Yes	Allowed to perform line measurements.
MeasurePoint	Yes	Allowed to perform point measurements.
MeasurePolygon	Yes	Allowed to perform polygon measurements.
MeasureHeight	Yes	Allowed to perform height measurements with the height tool.
MeasureVolume	Yes	Allowed to perform volume measurements with volume tool.
MeasureArea	Yes	Allowed to perform area measurement with area tool.

When calling a function on the API for which the user does not have sufficient rights, an exception is thrown.

2 Security sandboxes

2.1 Remote security sandbox

To allow using resources from a domain that does not host the loaded API SWF, host a crossdomain.xml at the root of that domain, having the following contents:

```
<cross-domain-policy>
    <site-control permitted-cross-domain-policies="all" />
    <allow-http-request-headers-from domain="*" headers="*" secure="false" />
    <allow-access-from domain="*" headers="*" secure="false" />
</cross-domain-policy>
```

The alternative is setting the useProxy flag to true when using various API functions. This assumes the hosted API can reach the proxy service.

2.2 Local security sandbox

To allow a local SWF the use of resources from locations other than that of the local file system, and to allow scripting between local HTML and SWF files, add a configuration file with references to the local files in either the global flash player trust directory, or the flash player trust directory of a specific user.

Example configuration file contents

```
D:\API training\Examples\api
D:\API training\Examples\demo
D:\API training\Examples\template
```

Global flash player trust (windows)

system\Macromed\Flash\FlashPlayerTrust

Example:
C:\WINDOWS\system32\Macromed\Flash\FlashPlayerTrust

User flash player trust (windows)

user application data\Macromedia\Flash Player\#Security\FlashPlayerTrust

Example:
C:\Documents and Settings\JohnD\Application Data\Macromedia\Flash Player\#Security\FlashPlayerTrust

Example:
C:\Users\JohnD\AppData\Roaming\Macromedia\Flash Player\#Security\FlashPlayerTrust

3 API initialization

This chapter only describes the initialization of the API. After initialization, all functions of the API will be available (only limited by the permissions granted based on the user credentials).

3.1 JavaScript Web and Microsoft .NET

Initialization is an important part in running an application developed using the API. This process is identical for all environments in which the application is run[1]. The initialization phase has two steps:

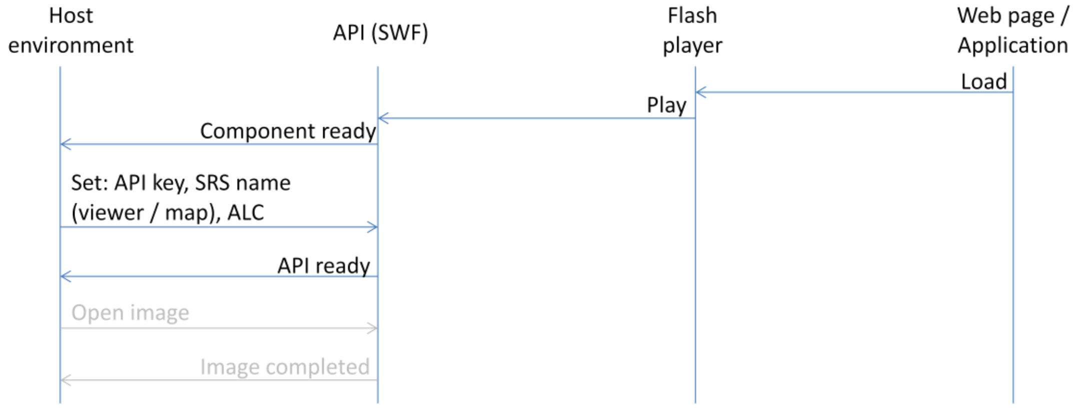
1. Wait until the Core API indicates to the host that it is ready for communication.
2. Set initialization parameters.

To indicate that it is possible to communicate with the Core API, it dispatches a component ready event. This practically means that it will attempt to call a function named `hst_componentReady`.

There are required and optional parameters that can only be set during the initialization phase. These parameters can only be entered during the `hst_componentReady` call. If all (required) parameters are entered and successfully validated, the Core API will dispatch the API ready event (by calling the `hst_apiReady` function). If one or more required parameters are not entered, or if one or more of the parameters cannot be successfully validated, the Core API will dispatch the API failed event (by calling the `hst_apiFailed` function).

After the API ready event has been dispatched all other functions are available (only limited by the permissions granted based on the user credentials).

The diagram below illustrates the initialization of the Core API.



The table below gives an overview of the parameters that can be entered during initialization (during the `hst_componentReady` call):

[1] Note that for a simple HTML integration, initialization is handled internally.

Parameter	*O/R*	*Function*	*Description*
API key	R	setAPIKey	A development key that references a system integrator.
Address language code	R	setAddressLanguageCode	The address language code (similar to a language locale) used for address lookups.
Set service URL	O	setServiceURL	The URLs to various services used by the API. The URLs have default values.
SRS name address	R*	setSRSNameAddress	A name that references a coordinate system. The GlobeSpotter API uses the EPSG format.
SRS name viewer	R**	setSRSNameViewer	A name that references a coordinate system. The GlobeSpotter API uses the EPSG format.
Temporary ID	O	setTID	A temporary ID for opening a cyclorama.
User credentials	O	setUserNamePassword	The name and password of the user. If entered, the user will not be confronted by a login box.

**A list of supported coordinate systems is supplied by the capabilities of the Cyclomedia recording location WFS: https://atlas.cyclomedia.com/Recordings/wfs?service=WFS&Request=GetCapabilities&Version=1.1.0

3.2 Simple Web HTML

HTML can be used to facilitate easy integration into static web pages or dynamically generated web pages. However, this form of integration only allows for a limited form of interaction. A HTML integration is most useful for simple viewing applications. The following example illustrates a HTML integration:

```
<html>
    <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
            id="viewer_api" width="800" height=400"
       <param name="movie" value="viewer_api" />
       <param name="allowScriptAccess" value="always" />
       <param name="allowFullScreen" value="true" />
       <embed src="https://globespotter.cyclomedia.com/v26/api/viewer_api.swf"
            quality="high"
            bgcolor="#888888"
            width="800" height="400"
            name="viewer_api"
            align="middle"
            play="true"
            loop="false"
            quality="high"
            allowScriptAccess="always"
            type="application/x-shockwave-flash"
            pluginspage="http://www.adobe.com/go/getflashplayer"
            allowFullScreen="true"
            flashvars="APIKey=abcdefghijklmnopqrstuvwxyz1234567890
               &SRSNameViewer=EPSG:28992&posx=145394&posy=427084&SRSNameAddress=EPSG:28992
               &AddressLanguageCode=nl"
            >
       </embed>
    </object>
</html>
```

The example above does not check if there is a flash player available and if its version matches the supported version. Parameters can be set using the **flashvars** property and are **not** case-sensitive. The table below gives an overview of the supported parameters:

Parameter	*O/R*	*Description*
address	O	The address that is used for opening a cyclorama.
apiKey	R	A development key that references a system integrator.
addressLanguageCode	R	The address language code (similar to a language locale) used for address lookups.
hFov	O	Sets the horizontal field of view. The value is in degrees and bounded by [30...130].
imageID	O	An image id used for opening a cyclorama.
pitch	O*	The vertical angle used for opening a cyclorama. The value is in degrees and bounded by [-90...90].
posX	O**	The x-element of a 2D coordinate for opening a cyclorama. The coordinate is referenced relative to the viewer SRS name.
posY	O**	The y-element of a 2D coordinate for opening a cyclorama. The coordinate is referenced relative to the viewer SRS name.
srsNameAddress	R	A name that references a coordinate system. The GlobeSpotter API uses the EPSG format.
srsNameViewer	R	A name that references a coordinate system. The GlobeSpotter API uses the EPSG format.
srsNameMap	O***	A name that references a coordinate system. The GlobeSpotter API uses the EPSG format.
tid		A temporary ID for opening a cyclorama.
yaw	O*	The horizontal angle used for opening a cyclorama. The value is in degrees and bounding by [0...360].

*Only applies when an image is opened using an image ID.
**The parameter *posX* cannot be referenced without *posY*.
*** It is only required when the full version of the API is used.

4 API compatibility

The following table shows all functionality with corresponding API version and an additional ‘deprecated since’ version to indicate in which version this particular functionality got deprecated.

Name	Introduced in API version	Deprecated since	Removed since	Renamed to
addAreaEntity	2.0.1024			
addGMLLayer	2.0.1024			
addHeightEntity	2.0.1024			
addLineMeasurement	2.0.1502			
addMapGMLLayer	2.0.1479		2.6.xxxx	addGMLLayer
addMapOSMLayer	2.0.1479		2.6.xxxx	setBaseLayer
addMapWFSLayer	2.0.1479		2.6.xxxx	addWFSLayer
addMapWMSLayer	2.0.1479		2.6.xxxx	addWMSLayer
addMeasurementPoint	2.7.xxxx			
addPointMeasurement	2.0.1502			
addSurfaceMeasurement	2.7.xxxx			
addVolumeEntity	2.0.1024			
addWFSLayer	2.0.1024			
addWMSLayer	2.6.xxxx			
applyStyle	2.6.xxxx			
autoComplete	2.8.xxxx			
cancelMeasurement	2.0.1502			
clearDrawing	2.0.1024		2.6.xxxx	clearMarker
clearDrawings	2.0.1024		2.6.xxxx	clearMarkers
clearMarker	2.6.xxxx			
clearMarkers	2.6.xxxx			
closeMap	2.0.1479		2.6.xxxx	
closeMeasurement	2.6.xxxx			
closeMeasurementPoint	2.7.xxxx			
closeViewer	2.6.xxxx			
createMeasurementPoint	2.7.xxxx			
drawMarkerAtHV	2.0.1024			

drawMarkerAtXY	2.0.1024			
drawMarkerInDirection	2.0.1024			
finishMeasurement	2.0.1502		2.6.xxxx	closeMeasurement
getAbsMaxViewers	2.0.1024		2.6.xxxx	A documented maximum of 5.
getActiveViewer	2.6.xxxx			
getActiveViewerReplaceMode	2.6.xxxx			
getAddressDatabase	2.7.xxxx			
getAddressLanguageCode	2.6.xxxx			
getAddressLocationsVisible	2.6.xxxx			
getAPIReadyState	2.0.1024			
getApplicationName	2.0.1024		2.6.xxxx	
getAutoMaximizeMap	2.6.xxxx			
getAutoPointPlacement	2.6.xxxx		2.7.xxxx	
getAutoTileViewers	2.0.1024	2.0.1479	2.6.xxxx	
getBrightness	2.6.xxxx			
getCloseViewerEnabled	2.6.xxxx			
getContextMenuEnabled	2.6.xxxx			
getDateFormat	2.8.xxxx			
getDateFrom	2.6.xxxx			
getDateTo	2.6.xxxx			
getDividerDrag	2.7.xxxx			
getDividerPosition	2.7.xxxx			
getDrawingLayerVisible	2.6.xxxx			
getEntityData	2.0.1024			
getEntityDescription	2.0.1024			
getEntityName	2.0.1024			
getFocusEntity	2.0.1024			
getFocusMode	2.6.xxxx			
getFocusPoint	2.6.xxxx			
getGamma	2.0.1024		2.6.xxxx	getBrightness
getGlobalViewerBrightness	2.7.xxxx			
getGlobalViewerOverlayAlpha	2.7.xxxx			
getGlobalViewerOverlayDrawDistance	2.7.xxxx			
getHFov	2.0.1024			
getHideOverlaysWhenMeasuring	2.6.xxxx			
getImageID	2.0.1024			
getImageInformationEnabled	2.6.xxxx			
getKeyboardEnabled	2.6.xxxx			
getLanguageLocale	2.0.1024			
getLogXML	2.6.xxxx			

getMajorVersion	2.0.1024			
getMapBorderColor	2.6.xxxx			
getMapCenter	2.0.1479			
getMapClickMode	2.0.1479			
getMapCompassVisible	2.6.xxxx			
getMapCoordinateInfoVisible	2.7.xxxx			
getMapCycleZoomLevelsEnabled	2.6.xxxx			
getMapEnabled	2.6.xxxx			
getMapExtent	2.0.1479			
getMapOverlayAlpha	2.7.xxxx			
getMapPrintImageEnabled	2.6.xxxx			
getMapRotation	2.6.xxxx			
getMapRotationEnabled	2.7.xxxx			
getMapSaveImageEnabled	2.6.xxxx			
getMapScaleLineVisible	2.7.xxxx			
getMapScreenshot	2.0.1479			
getMapState	2.0.1479		2.6.xxxx	
getMapTitleBarVisible	2.6.xxxx			
getMapTitleVisible	2.6.xxxx			
getMapToolBarButtonsVisible	2.7.xxxx			
getMapToolBarVisible	2.6.xxxx			
getMapWindowBorderVisible	2.6.xxxx			
getMapZoom	2.0.1479			
getMapZoomControlVisible	2.7.xxxx			
getMaxViewers	2.0.1024			
getMDICanvasDimensions	2.0.1479		2.6.xxxx	
getMDIWindowingMode	2.0.1479		2.6.xxxx	getWindowingMode
getMeasureLayerVisible	2.6.xxxx			
getMeasurementAnglesVisible	2.7.xxxx			
getMeasurementAreaVisible	2.7.xxxx			
getMeasurementDirection	2.7.xxxx			
getMeasurementDirectionEnabled	2.7.xxxx			
getMeasurementDistancesVisible	2.7.xxxx			
getMeasurementExtrusion	2.7.xxxx			
getMeasurementExtrusionEnabled	2.7.xxxx			
getMeasurementIDs	2.7.xxxx			
getMeasurementPointData	2.7.xxxx			
getMeasurementPointIDs	2.7.xxxx			
getMeasurementPointIndex	2.7.xxxx			
getMeasurementPointObservationData	2.7.xxxx			

getMeasurementPointObservationImageIDs	2.7.xxxx			
getMeasurementSeriesModeEnabled	2.7.xxxx			
getMeasurementSlopesVisible	2.7.xxxx			
getMinorVersion	2.0.1024			
getMeasurementSmartClickModeEnabled	2.8.xxxx			
getMouseInteractionEnabled	2.7.xxxx			
getOverlayAlpha	2.7.xxxx			
getOverlayDistanceEnabled	2.6.xxxx			
getOverlayDrawDistance	2.6.xxxx			
getPermissions	2.0.1024			
getPitch	2.0.1024			
getRecordingLocation	2.0.1024			
getRecordingLocationColorFromDate	2.7.xxxx			
getRecordingLocationsVisible	2.6.xxxx			
getRecycleFirstViewer	2.0.1024		2.6.xxxx	getActiveViewerReplaceMode
getShowMap	2.6.xxxx			
getSupportedAddressDatabases	2.7.xxxx			
getSupportedLanguageLocales	2.0.1024			
getSwapViewersEnabled	2.7.xxxx			
getTimeFormat	2.8.xxxx			
getUseDateRange	2.6.xxxx			
getViewerBorderColor	2.6.xxxx			
getViewerBrightnessEnabled	2.6.xxxx			
getViewerClickMode	2.0.1024			
getViewerCompassVisible	2.6.xxxx			
getViewerCount	2.0.1024			
getViewerCycleZoomLevelsEnabled	2.6.xxxx			
getViewerDetailImages	2.6.xxxx			
getViewerPrintImageEnabled	2.6.xxxx			
getViewerIDs	2.7.xxxx			
getViewerOverlayAlphaEnabled	2.6.xxxx			
getViewerOverlayDrawDistanceEnabled	2.6.xxxx			
getViewerRotationButtonsVisible	2.6.xxxx			
getViewerRotationEnabled	2.7.xxxx			
getViewerSaveImageEnabled	2.6.xxxx			
getViewerScreenshot	2.0.1479			
getViewerShowLocationEnabled	2.6.xxxx			

getViewerState	2.0.1479		2.6.xxxx	
getViewerTitleVisible	2.6.xxxx			
getViewerTitleBarVisible	2.6.xxxx			
getViewerToolBarVisible	2.6.xxxx			
getViewerToolBarButtonsVisible	2.7.xxxx			
getViewerWindowBorderVisible	2.0.1479			
getViewerWindowColor	2.0.1024			getViewerBorderColor
getViewerZoomBoxEnabled	2.0.1024			
getViewshotData	2.0.1479	2.6.xxxx	2.6.xxxx	
getTilingEnabled	2.6.xxxx			
getUnitAreaFactor	2.8.3912			
getUnitAreaLabel	2.8.3912			
getUnitAreaPrecision	2.8.3912			
getUnitLengthFactor	2.8.3912			
getUnitLengthLabel	2.8.3912			
getUnitLengthPrecision	2.8.3912			
getUnitVolumeFactor	2.8.3912			
getUnitVolumeLabel	2.8.3912			
getUnitVolumePrecision	2.8.3912			
getWindowingMode	2.0.1024			
getYaw	2.0.1024			
hideDrawingLayer	2.0.1024		2.6.xxxx	setDrawingLayerVisible
hideEntityLayer	2.0.1479		2.6.xxxx	setMeasureLayerVisible
hideMapEntityLayer	2.0.1479		2.6.xxxx	setMeasureLayerVisible
hideMapRLSLayer	2.0.1024		2.6.xxxx	setRecordingLocationsVisible
hideRecordingLocations	2.0.1024		2.6.xxxx	setRecordingLocationsVisible
hst_apiFailed	2.0.1024			
hst_apiReady	2.0.1024			
hst_autoCompleteResult	2.8.xxxx			
hst_componentReady	2.0.1024			
hst_detailImagesVisibilityChanged	2.8.3912			
hst_dividerPositionChanged	2.7.xxxx			
hst_entityDataChanged	2.0.1024			
hst_entityFocusChanged	2.6.xxxx			
hst_featureClicked	2.7.xxxx			
hst_focusPointChanged	2.6.xxxx			
hst_imageChanged	2.0.1024			
hst_imageCompleted	2.0.1024			
hst_imageFailed	2.0.1024			
hst_imagePreviewCompleted	2.0.1024			

hst_imageSegmentLoaded	2.0.1479			
hst_mapAdded	2.0.1479		2.6.xxxx	
hst_mapClicked	2.0.1479			
hst_mapExtentChanged	2.0.1479			
hst_mapInitialized	2.8.xxxx			
hst_mapRemoved	2.0.1479		2.6.xxxx	
hst_mapWindowMaximized	2.0.1479		2.6.xxxx	
hst_mapWindowMinimized	2.0.1479		2.6.xxxx	
hst_mapWindowMoved	2.0.1479		2.6.xxxx	
hst_mapWindowResized	2.0.1479		2.6.xxxx	
hst_mapWindowRestored	2.0.1479		2.6.xxxx	
hst_mapWindowSelected	2.0.1024		2.6.xxxx	
hst_markerClicked	2.6.xxxx			
hst_maxViewers	2.0.1479			
hst_measurementCanceled	2.6.xxxx			
hst_measurementClosed	2.0.1479			
hst_measurementCreated	2.0.1479			
hst_measurementFinished	2.0.1479		2.6.xxxx	hst_measurementClosed
hst_measurementModeChanged	2.6.xxxx			
hst_measurementOpened	2.0.1479			
hst_measurementPointAdded	2.7.xxxx			
hst_measurementPointClosed	2.7.xxxx			
hst_measurementPointObservation Added	2.7.xxxx			
hst_measurementPointObservation Removed	2.7.xxxx			
hst_measurementPointObservation Updated	2.7.xxxx			
hst_measurementPointOpened	2.7.xxxx			
hst_measurementPointRemoved	2.7.xxxx			
hst_measurementPointUpdated	2.7.xxxx			
hst_measurementUpdated	2.0.1479	2.7.xxxx	2.7.xxxx	
hst_observationAdded	2.0.1479		2.7.xxxx	hst_measurementPointObservation Added
hst_observationRemoved	2.0.1479		2.7.xxxx	hst_measurementPointObservation Removed
hst_observationUpdated	2.0.1024		2.7.xxxx	hst_measurementPointObservation Updated
hst_openImageError	2.6.xxxx		2.6.xxxx	hst_openImageFailed
hst_openImageFailed	2.6.xxxx			
hst_openImageResult	2.0.1024			
hst_openImageResultEmpty	2.6.xxxx		2.6.xxxx	hst_openImageResult, hst_openNearestImageResult

hst_openNearestImageResult	2.0.1024			
hst_showLocationRequested	2.8.xxxx			
hst_viewChanged	2.6.xxxx			
hst_viewClicked	2.6.xxxx			
hst_viewerActive	2.0.1024			
hst_viewerAdded	2.0.1024			
hst_viewerClicked	2.6.xxxx		2.6.xxxx	hst_viewClicked
hst_viewerInactive	2.0.1024			
hst_viewerRemoved	2.0.1479			
hst_viewerWindowMaximized	2.0.1479		2.6.xxxx	
hst_viewerWindowMinimized	2.0.1479		2.6.xxxx	
hst_viewerWindowMoved	2.0.1479		2.6.xxxx	
hst_viewerWindowResized	2.0.1479		2.6.xxxx	
hst_viewerWindowRestored	2.0.1479		2.6.xxxx	
hst_viewerWindowSelected	2.0.1024		2.6.xxxx	
hst_viewLoaded	2.6.xxxx			
hst_wfsFeatureClicked	2.0.1024		2.7.xxxx	hst_featureClicked
lookAtCoordinate	2.0.1479			
maximizeMapWindow	2.0.1479		2.6.xxxx	
maximizeViewerWindow	2.0.1479		2.6.xxxx	
minimizeMapWindow	2.0.1479		2.6.xxxx	
minimizeViewerWindow	2.0.1479		2.6.xxxx	
moveMapWindow	2.0.1479		2.6.xxxx	
moveMeasurementPointToIndex	2.7.xxxx			
moveViewerWindow	2.0.1024		2.6.xxxx	
nextMeasurementSeries	2.7.xxxx			
openImage	2.0.1479			
openMap	2.6.xxxx		2.7.xxxx	setBaseLayer(), setMapEnabled(), setShowMap ()
openMeasurement	2.6.xxxx			
openMeasurementPoint	2.7.xxxx			
openNearestImage	2.0.1024			
removeAllEntities	2.0.1024			
removeEntity	2.0.1024			
removeGMLLayer	2.6.xxxx		2.6.xxxx	removeLayer
removeLayer	2.0.1024			
removeMapLayer	2.0.1024		2.6.xxxx	removeLayer
removeMeasurementPoint	2.7.xxxx			
removeMeasurementPointObservation	2.7.xxxx			
removeViewer	2.0.1024		2.6.xxxx	closeViewer

removeWFSLayer	2.0.1479		2.6.xxxx	removeLayer
resetMeasurementSeries	2.7.xxxx			
resizeMapWindow	2.0.1479		2.6.xxxx	
resizeViewerWindow	2.0.1479		2.6.xxxx	
restoreMapWindow	2.0.1479		2.6.xxxx	
restoreViewerWindow	2.0.1024		2.6.xxxx	
rotateDown	2.0.1024			
rotateLeft	2.0.1024			
rotateRight	2.0.1024			
rotateUp	2.0.1479			
selectMapWindow	2.0.1479		2.6.xxxx	
selectViewerWindow	2.6.xxxx		2.6.xxxx	setActiveViewer
setActiveViewer	2.6.xxxx			
setActiveViewerReplaceMode	2.6.xxxx			
setAddressLanguageCode	2.6.xxxx			
setAddressLocationsVisible	2.6.xxxx			
setAPIKey	2.0.1024			
setApplicationParameter	2.6.xxxx		2.6.xxxx	setAPIKey, setSRSName..., setServiceURL, setTID
setAutoMaximizeMap	2.6.xxxx			
setAutoPointPlacement	2.6.xxxx		2.7.xxxx	
setAutoTileViewers	2.0.1479	2.6.xxxx	2.6.xxxx	
setBaseLayer	2.6.xxxx			
setBrightness	2.6.xxxx			
setCloseViewerEnabled	2.6.xxxx			
setContextMenuEnabled	2.6.xxxx			
setDateFormat	2.8.xxxx			
setDateFrom	2.6.xxxx			
setDateTo	2.6.xxxx			
setDividerDrag	2.7.xxxx			
setDividerPosition	2.7.xxxx			
setDrawingColor	2.0.1024		2.6.xxxx	setMarkerColor
setDrawingImageURL	2.6.xxxx		2.6.xxxx	setMarkerImageURL
setDrawingLayerVisible	2.0.1024			
setDrawingMode	2.0.1024			
setDrawingSize	2.0.1024		2.6.xxxx	setMarkerSize
setEntityDescription	2.0.1024			
setFocusEntity	2.6.xxxx			
setFocusMode	2.6.xxxx			
setFocusPoint	2.0.1024			
setGamma	2.0.1024		2.6.xxxx	setBrightness

setGlobalViewerBrightness	2.7.xxxx			
setGlobalViewerOverlayAlpha	2.7.xxxx			
setGlobalViewerOverlayDrawDistance	2.7.xxxx			
setHFov	2.6.xxxx			
setHideOverlaysWhenMeasuring	2.6.xxxx			
setImageInformationEnabled	2.6.xxxx			
setKeyboardEnabled	2.0.1024			
setLanguageLocale	2.6.xxxx			
setMapBorderColor	2.0.1479		2.6.xxxx	
setMapCenter	2.0.1479			
setMapClickMode	2.6.xxxx			
setMapCompassVisible	2.6.xxxx			
setMapCoordinateInfoVisible	2.7.xxxx			
setMapCycleZoomLevelsEnabled	2.6.xxxx			
setMapEnabled	2.0.1479			
setMapExtent	2.0.1479			
setMapOverlayAlpha	2.7.xxxx			
setMapPrintImageEnabled	2.6.xxxx			
setMapRLSLayerDateRange	2.0.1479		2.6.xxxx	setDateFrom, getDateFrom, setDateTo, getDateTo
setMapRotation	2.6.xxxx			
setMapRotationEnabled	2.7.xxxx			
setMapSaveImageEnabled	2.6.xxxx			
setMapScaleLineVisible	2.7.xxxx			
setMapTitleBarVisible	2.6.xxxx			
setMapTitleVisible	2.6.xxxx			
setMapToolBarButtonsVisible	2.7.xxxx			
setMapToolBarVisible	2.6.xxxx			
setMapWindowBorderVisible	2.6.xxxx			
setMapZoom	2.6.xxxx			
setMapZoomControlVisible	2.7.xxxx			
setMarkerColor	2.6.xxxx			
setMarkerImageURL	2.6.xxxx			
setMarkerSize	2.0.1024			
setMaxViewers	2.0.1479			
setMDIWindowingMode	2.6.xxxx		2.6.xxxx	setWindowingMode
setMeasurementAnglesVisible	2.7.xxxx			
setMeasurementAreaVisible	2.7.xxxx			
setMeasurementDirection	2.7.xxxx			
setMeasurementDirectionEnabled	2.7.xxxx			

setMeasurementExtrusion	2.7.xxxx			
setMeasurementExtrusionEnabled	2.7.xxxx			
setMeasurementDistancesVisible	2.7.xxxx			
setMeasurementSeriesModeEnabled	2.7.xxxx			
setMeasurementSlopesVisible	2.7.xxxx			
setMeasurementSmartClickModeEnabled	2.8.xxxx			
setMeasureLayerVisible	2.6.xxxx			
setMouseInteractionEnabled	2.7.xxxx			
setOverlayAlpha	2.6.xxxx			
setOverlayDrawDistance	2.6.xxxx			
setPitch	2.6.xxxx			
setRecordingLocationsVisible	2.6.xxxx			
setRecycleFirstViewer	2.0.1479		2.6.xxxx	setActiveViewerReplaceMode
setServiceURL	2.6.xxxx			
setShowMap	2.6.xxxx			
setSRSNameAddress	2.6.xxxx			
setSRSNameViewer	2.0.1479			
setSwapViewersEnabled	2.7.xxxx			
setTID	2.6.xxxx			
setTimeFormat	2.8.xxxx			
setUnitAreaFactor	2.8.3912			
setUnitAreaLabel	2.8.3912			
setUnitAreaPrecision	2.8.3912			
setUnitLengthFactor	2.8.3912			
setUnitLengthLabel	2.8.3912			
setUnitLengthPrecision	2.8.3912			
setUnitVolumeFactor	2.8.3912			
setUnitVolumeLabel	2.8.3912			
setUnitVolumePrecision	2.8.3912			
setUseDateRange	2.0.1024			
setUserNamePassword	2.6.xxxx			
setViewerBorderColorScheme	2.6.xxxx			
setViewerBrightnessEnabled	2.6.xxxx			
setViewerClickMode	2.6.xxxx			
setViewerCompassVisible	2.6.xxxx			
setViewerCompassVisible	2.6.xxxx			
setViewerCycleZoomLevelsEnabled	2.6.xxxx			
setViewerDetailImagesVisible	2.6.xxxx			
setViewerOverlayAlphaEnabled	2.6.xxxx			

setViewerOverlayDrawDistanceEnabled	2.6.xxxx			
setViewerPrintImageEnabled	2.6.xxxx			
setViewerRotationButtonsVisible	2.6.xxxx			
setViewerRotationEnabled	2.7.xxxx			
setViewerSaveImageEnabled	2.6.xxxx			
setViewerShowLocationEnabled	2.6.xxxx			
setViewerTitleVisible	2.6.xxxx			
setViewerTitleBarVisible	2.6.xxxx			
setViewerToolBarVisible	2.6.xxxx			
setViewerToolBarButtonsVisible	2.7.xxxx			
setViewerWindowBorderVisible	2.6.xxxx			
setViewerZoomBoxEnabled	2.6.xxxx			
setTilingEnabled	2.6.xxxx			
setWindowingMode	2.6.xxxx			
setYaw	2.0.1024			
showDrawingLayer	2.0.1024		2.6.xxxx	setDrawingLayerVisible
showEntityLayer	2.6.xxxx		2.6.xxxx	setMeasureLayerVisible
showImageInformation	2.0.1479			
showMapEntityLayer	2.0.1479		2.6.xxxx	setMeasureLayerVisible
showMapPrintDialog	2.0.1479			
showMapRLSLayer	2.0.1479		2.6.xxxx	setRecordingLocationsVisible
showMapSaveDialog	2.0.1024			
showRecordingLocations	2.6.xxxx		2.6.xxxx	setRecordingLocationsVisible
showViewerLocationOnMap	2.0.1479			
showViewerPrintDialog	2.0.1479			
showViewerSaveDialog	2.0.1024			
takeViewshot	2.6.xxxx	2.0.1479	2.6.xxxx	
zoomViewerToMaxLevel	2.6.xxxx			
zoomViewerToMedLevel	2.6.xxxx			
zoomViewerToMinLevel	2.6.xxxx			

5 API reference list

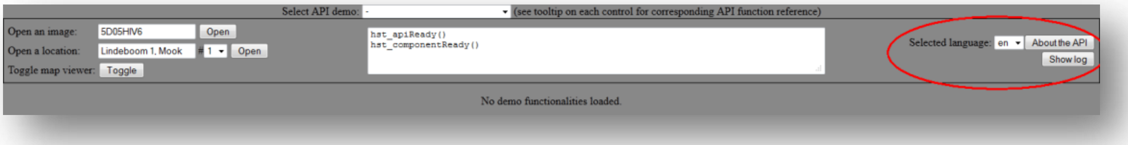
This chapter details the various functions and events that are available through the API.

Functions can be called by the host via the API object. Events are dispatched by the API and attempt to run a call-back function that may or may not be made available by the host. It is important to note, that within the API, events that run call-backs functions are wrapped inside a try-catch block. This means that any exceptions that occur within the call-back function are silently ignored, and can only be detected if the content of the call-back function that throws the exception is wrapped within a try-catch block.

In case of *getter* and *setter* functions, the *setter* functions will usually contain a detailed description of its use (which also applies to the *getter*).

5.1 General

This section describes the various functions and events that handle general API properties and initialization.



Select API demo: - (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

Toggle map viewer:

```
hst_apiReady()
hst_componentReady()
```

Selected language: en

No demo functionalities loaded.

The demo for this section can be found on the general demo page. Corresponding example source code can be found in index.html.

5.1.1 Functions

5.1.1.1 getAddressDatabase

Returns the database used for address related image searches.

Returns	Type	Description
name	String	The name of the database

5.1.1.2 getAddressLanguageCode

Returns the language code used for address related image searches.

Returns	Type	Description
code	String	A language code.

5.1.1.3 getAPIReadyState

Returns if the API can be used for general use.

Returns	Type	Description
readyState	Boolean	The ready state of the API.

5.1.1.4 getLanguageLocale

Returns the locale of the language used.

Returns	Type	Description
locale	String	The locale of the language.

5.1.1.5 getLogXML

Returns the log in XML form.

Returns	Type	Description
log	String	The log as an XML string.

5.1.1.6 getMajorVersion

Returns the major release version of the current application. It indicates global changes within the application and can be used for debugging purposes. Global changes may indicate a change in the interface of the application.

Returns	Type	Description
version	String	The major release version

5.1.1.7 getMinorVersion

Returns the minor release version of the current application. It indicates minor changes within the application and can be used for debugging purposes (specify bugs).

Returns	Type	Description
version	String	The minor release version

5.1.1.8 getPermissions

Retrieves the permissions of the user based on the users credentials.

Returns	Type	Description
permissions	Object	Permissions of the user based on the users credentials.

5.1.1.9 getSupportedAddressDatabases

Retrieves the names of all supported address databases.

Returns	Type	Description
names	String[]	The names of the supported address databases.

5.1.1.10 getSupportedLanguageLocales

Retrieves all supported language-locales.

Returns	Type	Description
locales	String[]	All supported locales.

5.1.1.11 setAddressDatabase

Sets the database to be used for address lookups. Supported names can be retrieved using *getSupportedAddressDatabases*.

Parameter	Type	Description
name	String	The name of the database

5.1.1.12 setAddressLanguageCode

The address language code determines the database used for address lookups.

It is required when initializing the API and results in an API failed if an invalid code is set. It can be changed even after the API has been initialized, potentially allowing lookups for multiple countries. Examples are nl, de, be, cz, pl, etc.

Parameter	Type	Description
code	String	Language code

5.1.1.13 setAPIKey

Sets the development key that references a system integrator.

Parameter	Type	Description
apiKey	String	Developer key.

5.1.1.14 setLanguageLocale

Sets the current language-locale.

Parameter	Type	Description
locale	String	The locale of the language.

Returns	Type	Description
isLocaleSupported	Boolean	If the locale is supported.

5.1.1.15 setServiceURL

It is possible for our customers to host their images on local atlas service therefore the atlas service URL's should be configurable.

The default urls point to https://atlas.cyclomedia.com

Default service parameter for setting the base URL to local services:

BASE_URL = 0x100 /* http://my.servicelocation.com/ */

Services can be set individually as well, but this is not recommended.

URL_ADDRESS_SERVICE = 0x101
URL_PANORAMA_SERVICE = 0x102
URL_RECORDING_LOCATION_SERVICE = 0x103
URL_OBLIQUE_LOCATION_SERVICE = 0x107
URL_OBLIQUE_TILING_SERVICE = 0x108
URL_SMART_CLICK_SERVICE = 0x106

A local installation is available for testing purposes on
http://localinstallation.globespotter.nl
username/password for this installation is cyclotest/cyclotest

The 2.8 api on this server is hosted at :
http://localinstallation.globespotter.nl/globespotter/2.8/api/viewer_api.swf

A small part of the city of Eindhoven is available on this server. Use the following link for the location:
http://localinstallation.globespotter.nl/globespotter/2.8/viewer/index.html?address=markt,eindhoven

Parameter	Type	Description
service	uint	The type of service.
url	String	The URL of the service.

5.1.1.16 setSRSNameAddress

Set the address spatial reference system using an srs name (e.g. EPSG:28992).

Parameter	Type	Description
srsName	String	The srs name in EPSG format.

5.1.1.17 setSRSNameViewer

Set the viewer spatial reference system using an srs name (e.g. EPSG:28992).

Parameter	Type	Description
srsName	String	The srs name in EPSG format.

5.1.1.18 setTID

Sets the temporary identification string that will be used for authentication of the requested images and recordinglocations.

Parameter	Type	Description
TID	String	Temporary identification string.

5.1.1.19 setUserNamePassword

Sets the credentials of the user. When successful, the user will not be asked to provide his or her credentials before (almost) each action.

Parameter	Type	Description
userName	String	The name or ID of the user.
password	String	The password that belongs to the user-name or ID.

5.1.2 Events

5.1.2.1 hst_apiFailed

Indicates to the host that the API failed to initialize.

Parameter	Type	Description

5.1.2.2 hst_apiReady

Indicates to the host that the API is ready for use. Full API functionality should be available based on the user's credentials.

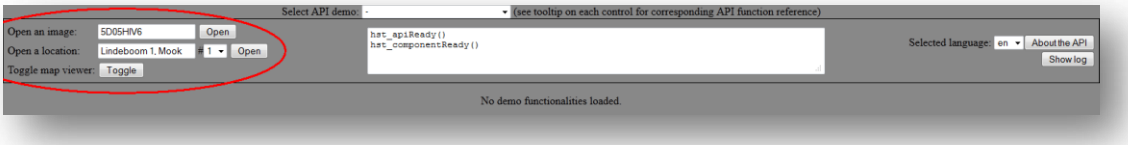
Parameter	Type	Description

5.1.2.3 *hst_componentReady*

Indicates to the host that the component is ready for use. After this event the user is able to set various required application parameters to be able to use the components API functionality.

Parameter	Type	Description

5.2 Opening images



Select API demo: - (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

Toggle map viewer:

```
hst_apiReady()
hst_componentReady()
```

Selected language: en

No demo functionalities loaded.

The demo for this section can be found on the general demo page. Corresponding example source code can be found in index.html.

5.2.1.1 *autoComplete*

This function auto-completes a search term by returning a list of possible matches (see *hst_autoCompleteResult)*.Result can then be opened using *openNearestImage*.

Parameter	Type	Description
term	String	The term to autocomplete

5.2.1.2 *openImage*

This function attempts to open an image using an image id. An initial yaw, pitch and horizontal field of view can be set using a parameter object having the following properties:

- yaw [float] in degrees
- pitch [float] in degrees
- hFov [float] in degrees
- dateFrom [String] the minimum date for recordings as an ISO 8601 DateTime string (optional)
- dateTo [String] the maximum date for recordings as an ISO 8601 DateTime string (optional)

Parameter	Type	Description
imageID	String	
viewerParams	Object	(Optional) Default = null. Details about how to open an image.

5.2.1.3 openNearestImage

This function attempts to open an image via a coordinate or address.

Parameter	Type	Description
value	String	The request coordinate or address
count	Int	Amount of nearest viewers to be opened.

5.2.2 Events

5.2.2.1 hst_autoCompleteResult

Indicates to the host the result of an auto-complete request. The first entry in the list represents the closest match.

Parameter	Type	Description
results	Array[String]	A list of possible matches

5.2.2.2 hst_openImageFailed

Indicates to the host that the API was unable to open an image due to an error generated during the opening process. The error is most likely caused by one of the following reasons:

- Web service timeout
- Insufficient rights to use the services

Parameter	Type	Description
request	String	Search request string for the openImage or openNearestImage function.

5.2.2.3 hst_openImageResult

Indicates to the host the result of an open image call. If the API was unable to open an image, then the image ID was not found.

Parameter	Type	Description
request	String	‘value’ parameter from previously called openImage
opened	Boolean	True if the API was able to open an image.
viewerID	uint	ID of viewer

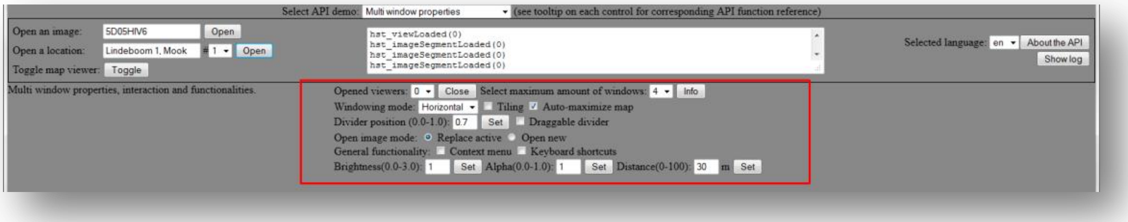
5.2.2.4 hst_openNearestImageResult

Indicates to the host the result of an open image call. If the API was unable to open an image, then any of the following problems may be valid:

- Address not found
- No image found near provided address
- No image found near provided X,Y location

Parameter	Type	Description
request	String	‘value’ parameter from previously called openNearestImage.
opened	Boolean	True if the API was able to open an image.
viewerID	uint	ID of viewer.

5.3 Multi window properties



Select API demo: Multi window properties (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

Toggle map viewer:

```
hst_viewLoaded(0)
hst_imageSegmentLoaded(0)
hst_imageSegmentLoaded(0)
hst_imageSegmentLoaded(0)
```

Selected language: en

Multi window properties, interaction and functionalities.

Opened viewers: 0 Select maximum amount of windows: 4

Windowing mode: Horizontal □ Tiling ☑ Auto-maximize map

Divider position (0.0-1.0): 0.7 □ Draggable divider

Open image mode: ● Replace active ○ Open new

General functionality: □ Context menu □ Keyboard shortcuts

Brightness(0.0-3.0): 1 Alpha(0.0-1.0): 1 Distance(0-100): 30 m

The demo for this section can be found on the ‘Multi window properties’ demo page. Corresponding example source code can be found in japi/demo-multi-window.html.

5.3.1 Functions

5.3.1.1 closeViewer

Remove a viewer from the window manager using a viewer id.

Parameter	Type	Description
viewerID	uint	A reference to a viewer.

5.3.1.2 getActiveViewer

Gets which viewer-window is active.

Returns	Type	Description
viewerID	int	A reference to the active viewer. An id of -1 is returned if no viewer is active.

5.3.1.3 getActiveViewerReplaceMode

Returns true if the active viewer is replaced when opening an image.

Returns	Type	Description
replaceActiveViewer	Boolean	Replace mode enabled.

5.3.1.4 getAutoMaximizeMap

Returns true if the map is maximized when no viewer is opened.

Returns	Type	Description
mapAutoMaximized	Boolean	

5.3.1.5 getDateFormat

Returns the current format used for date visualisation.

Dates and times are formatted using the following definition:
http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/mx/formatters/DateFormatter.html#formatString

Returns	Type	Description
format	String	

5.3.1.6 getDividerDrag

Returns true if the divider between the viewer groups can be dragged.

Returns	Type	Description
mapAutoMaximized	Boolean	

5.3.1.7 getDividerPosition

Returns the current position of the divider between the viewer groups.
Returns a value between 0.25 (maximum Cyclorama viewer) and 0.75 (maximum map viewer).

Returns	Type	Description
position	float	The current setting.

5.3.1.8 getContextMenuEnabled

Returns true if context menu is enabled for the window manager.

Returns	Type	Description
enabled	Boolean	

5.3.1.9 getGlobalViewerBrightness

Returns the global brightness for all Cyclorama viewers.

Returns	Type	Description
brightness	float	A brightness value within the range of [0.1..3.0]

5.3.1.10 getGlobalViewerOverlayAlpha

Returns the global overlay alpha for all Cyclorama viewers.

Returns	Type	Description
alpha	float	An alpha value within the range of [0.0..1.0]

5.3.1.11 getGlobalViewerOverlayDrawDistance

Returns the global overlay draw distance for all Cyclorama viewers.

Returns	Type	Description
drawDistance	float	An draw distance value within the range of [0.0..100.0]

5.3.1.12 getKeyboardEnabled

Returns true if keyboard shortcuts are available for use.

Returns	Type	Description
enabled	Boolean	

5.3.1.13 getMapEnabled

Returns true if the map is enabled.

Returns	Type	Description
enabled	Boolean	

5.3.1.14 getMaxViewers

Returns the maximum number of viewers that can be opened by the viewer manager.

Returns	Type	Description
maxViewers	uint	The maximum number of viewers that can be opened.

5.3.1.15 getMouseInteractionEnabled

Returns true if mouse interaction is enabled for the viewer camera.

Returns	Type	Description
enabled	Boolean	

5.3.1.16 getTilingEnabled

Returns true if tiling is enabled.

Returns	Type	Description
enabled	Boolean	

5.3.1.17 getTimeFormat

Returns the current format used for time visualisation.

Dates and times are formatted using the following definition:
http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/mx/formatters/DateFormatter.html#formatString

Returns	Type	Description
format	String	

5.3.1.18 getShowMap

Returns true if the map viewer is visible.

Returns	Type	Description
visible	Boolean	

5.3.1.19 getUnitAreaFactor

Returns the current factor used for presentation of the area units.

Returns	Type	Description
factor	float	Determines how many meters go into the unit to be used (by default UnitLengthFactor * UnitLenghtFactor will be used).

5.3.1.20 getUnitAreaLabel

Returns the current label used for presentation of the area units.

Returns	Type	Description
label	String	

5.3.1.21 getUnitAreaPrecision

Returns the current precision used for presentation of the area units.

Returns	Type	Description
precision	float	

5.3.1.22 getUnitLengthFactor

Returns the current factor used for presentation of the length units.

Returns	Type	Description
factor	float	

5.3.1.23 getUnitLengthLabel

Returns the current label used for presentation of the length units.

Returns	Type	Description
label	String	

5.3.1.24 getUnitLengthPrecision

Returns the current precision used for presentation of the length units.

Returns	Type	Description
precision	float	

5.3.1.25 getUnitVolumeFactor

Returns the current factor used for presentation of the volume units.

Returns	Type	Description
factor	float	

5.3.1.26 getUnitVolumeLabel

Returns the current label used for presentation of the volume units.

Returns	Type	Description
label	String	

5.3.1.27 getUnitVolumePrecision

Returns the current precision used for presentation of the volume units.

Returns	Type	Description
precision	float	

5.3.1.28 getViewerCount

Get the number of viewers that are currently opened within the viewer manager.

Returns	Type	Description
viewerCount	uint	The number of viewers opened.

5.3.1.29 getViewerIDs

Return the list of current viewer IDs.

Returns	Type	Description

list	Array	The list of IDs [int] of the Cyclorama viewer currently opened

5.3.1.30 getWindowingMode

Returns the current windowing mode.

Returns	Type	Description
windowingMode	uint	

Returns	Type	Description
enabled	Boolean	

5.3.1.31 setContextMenuEnabled

If set to true, extra context menu items are made available within the window manager.

Parameter	Type	Description
enabled	Boolean	

5.3.1.32 setDateFormat

Sets the current format used for date visualisation.

Dates and times are formatted using the following definition: http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/mx/formatters/DateFormatter.html#formatString

Parameter	Type	Description
format	String	

5.3.1.33 setKeyboardEnabled

If set to true, keyboard shortcuts will be made available. These shortcuts are identical to those of GlobeSpotter. See the Globespotter manual for a description of all available shortcuts.

Parameter	Type	Description
enabled	Boolean	

5.3.1.34 setMapEnabled

Enables or disables the map. Set a base layer first using `setBaseLayer`. The map viewer will be visible if `getShowMap` equals `true`.

Parameter	Type	Description
enabled	Boolean	

5.3.1.35 setMouseInteractionEnabled

Enables/disables the mouse interaction for the viewer camera.

Parameter	Type	Description
enabled	Boolean	

5.3.1.36 *setShowMap*

Hides/show the map viewer (if it is enabled).

Parameter	Type	Description
visible	Boolean	

5.3.1.37 *setTimeFormat*

Sets the current format used for time visualisation.

Dates and times are formatted using the following definition:
http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/mx/formatters/DateFormatter.html#formatString

Parameter	Type	Description
format	String	

5.3.1.38 *setActiveViewer*

Makes a viewer active.

Parameter	Type	Description
viewerID	uint	A reference to a viewer.

5.3.1.39 *setActiveViewerReplaceMode*

If set to true, an active viewer is replaced when opening an image.

Parameter	Type	Description
replaceMode	Boolean	True if enabled.

5.3.1.40 *setAutoMaximizeMap*

If true, the map is maximized when no viewer is opened. Otherwise, the map does not fill the available room.

Parameter	Type	Description
autoMaximizeMap	Boolean	

5.3.1.41 *setDividerDrag*

Sets the divider between the viewer groups to draggable.

Parameter	Type	Description
dividerDrag	Boolean	

5.3.1.42 setDividerPosition

Sets the current position of the divider between the viewer groups.
Must be a value between 0.25 (maximum Map viewer) and 0.75 (maximum Cyclorama viewers).

Parameter	Type	Description
position	float	

5.3.1.43 setGlobalViewerBrightness

Sets the global brightness for all Cyclorama viewers.

Parameter	Type	Description
brightness	float	A brightness value within the range of [0.1..3.0]

5.3.1.44 setGlobalViewerOverlayAlpha

Sets the global overlay alpha for all Cyclorama viewers.

Parameter	Type	Description
alpha	float	An alpha value within the range of [0.0..1.0]

5.3.1.45 setGlobalViewerOverlayDrawDistance

Sets the global overlay draw distance for all Cyclorama viewers.

Parameter	Type	Description
drawDistance	float	An draw distance value within the range of [0.0..100.0]

5.3.1.46 setMaxViewers

Sets the maximum number of viewers (bounded by *[1...absMaxViewers]*). The absolute maximum is 6.

Parameter	Type	Description
maxViewers	uint	The requested maximum number of viewers.

Returns	Type	Description
maxViewers	uint	The bounded maximum number of viewers.

5.3.1.47 setTilingEnabled

Enables/disables tiling mode. In tiling mode the viewer layout will use two rows (in horizontal windowing mode) or two columns (in vertical windowing mode) when there is more than one viewer opened.

Parameter	Type	Description
tilingEnabled	Boolean	

5.3.1.48 setUnitAreaFactor

Sets the current factor used for presentation of the area units.

Parameter	Type	Description
factor	float	Determines how many m² go into the unit to be used (by default (UnitLengthFactor)² will be used).

5.3.1.49 setUnitAreaLabel

Sets the current label used for presentation of the area units.

Parameter	Type	Description
label	String	By default the UnitLengthLabel + ‘²’ is used.

5.3.1.50 setUnitAreaPrecision

Sets the current precision used for presentation of the area units.

Parameter	Type	Description
precision	float	By default a precision of 2 decimals is used.

5.3.1.51 setUnitLengthFactor

Sets the current factor used for presentation of the length units.

Parameter	Type	Description
factor	float	By default a value of 1.0 is used.

5.3.1.52 setUnitLengthLabel

Sets the current label used for presentation of the length units.

Parameter	Type	Description
label	String	By default ‘m’ is used.

5.3.1.53 setUnitLengthPrecision

Sets the current precision used for presentation of the length units.

Parameter	Type	Description
precision	float	By default a precision of 2 decimals is used.

5.3.1.54 setUnitVolumeFactor

Sets the current factor used for presentation of the volume units.

Parameter	Type	Description
factor	float	Determines how many m^3 go into the unit to be used (by default (UnitLengthFactor)^3 will be used).

5.3.1.55 setUnitVolumeLabel

Sets the current label used for presentation of the volume units.

Parameter	Type	Description
label	String	By default the UnitLengthLabel + '^{3}' is used.

5.3.1.56 setUnitVolumePrecision

Sets the current precision used for presentation of the volume units.

Parameter	Type	Description
precision	float	By default a precision of 2 decimals is used.

5.3.1.57 setWindowingMode

Sets the windowing mode. The selected mode determines the positioning behaviour of windows when they are added or removed.

Supported windowing modes:

```
Horizontal=1
Vertical=2
```

Parameter	Type	Description
windowingMode	uint	

5.3.2 Events

5.3.2.1 hst_dividerPositionChanged

Indicaties to the host that the position of the divider between viewergroups had been updated (the user dragged the divider).

Parameter	Type	Description
dividerPosition	float	The position of the divider [0.25...0.75], where 0.25 is indicates the maximum size for the map viewer and 0.75 the maximum size for the Cyclorama viewers.

5.3.2.2 hst_maxViewers

Indicates to the host that the maximum amount of viewers that can be opened is reached.

5.3.2.3 hst_viewerActive

Indicates to the host that a viewer is now active. This happens when a users selects a viewer.

Parameter	Type	Description
viewerID	uint	

5.3.2.4 hst_viewerAdded

Indicates to the host that a viewer was added. This happens when a viewer is successfully opened.

Parameter	Type	Description
viewerID	uint	

5.3.2.5 hst_viewerInactive

Indicates to the host that a viewer is now inactive. This happens when a users selects another viewer.

Parameter	Type	Description
viewerID	uint	

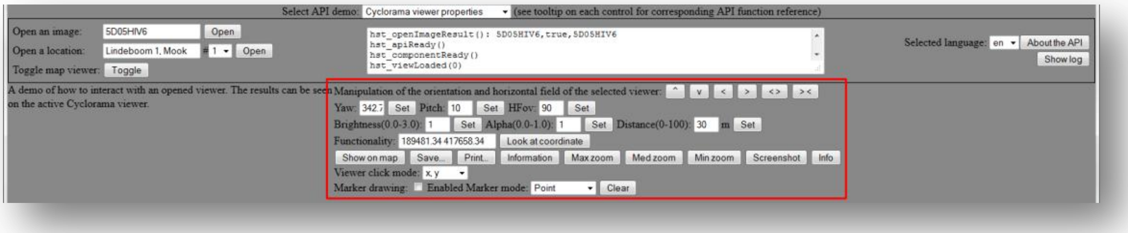
5.3.2.6 hst_viewerRemoved

Indicates to the host that a viewer was removed.

Parameter	Type	Description

viewerID	uint	

5.4 Cyclorama viewer properties



Select API demo: Cyclorama viewer properties (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

```
hst_openImageResult(): 5D05HIV6,true,5D05HIV6
hst_apiReady()
hst_componentReady()
hst_viewLoaded(0)
```

Selected language: en

A demo of how to interact with an opened viewer. The results can be seen on the active Cyclorama viewer.

Yaw: 342.7 Pitch: 10 HFov: 90

Brightness(0.0-3.0): 1 Alpha(0.0-1.0): 1 Distance(0-100): 30 m

Functionality: 189481.34 417658.34

Viewer click mode: x, y

Marker drawing: □ Enabled Marker mode: Point

The demo for this section can be found on the ‘Cyclorama viewer properties’ demo page. Corresponding example source code can be found in japi/demo-cyclorama-viewer-properties.html.

5.4.1 Functions

5.4.1.1 *getBrightness*

Gets the gamma of a viewer.

Parameter	Type	Description
viewerID	uint	The viewer id from which to obtain the data.

Returns	Type	Description
brightness	float	A brightness value within the range of [0.1..3.0]

5.4.1.2 *getDrawingLayerVisible*

Gets the visibility of the viewer drawing layer.

Parameter	Type	Description
Visible	Boolean	

5.4.1.3 *getHFov*

Gets the horizontal field of view in degrees.

Parameter	Type	Description
viewerID	uint	The viewer id from which to obtain the data.

Returns	Type	Description
hFov	float	The horizontal field of view in degrees.

5.4.1.4 *getImageID*

Gets the image id of a viewer.

Parameter	Type	Description
viewerID	uint	The viewer id from which to obtain the data.

Returns	Type	Description
imageID	String	The id of the image

5.4.1.5 getOverlayAlpha

Gets the alpha value of the overlays on a specific viewer.

Parameter	Type	Description
viewerID	uint	The viewer id from which to obtain the data.

Returns	Type	Description
alpha	float	The alpha values for the overlays [0.0...1.0]

5.4.1.6 getOverlayDrawDistance

Gets the distance, in meters, used for drawing layers from the camera center within a cyclorama.

Parameter	Type	Description
viewerID	uint	The viewer id from which to obtain the data.

Returns	Type	Description
distance	float	The draw distance for layers.

5.4.1.7 getPitch

Gets the vertical orientation of a view in degrees.

Parameter	Type	Description
viewerID	uint	The viewer id from which to obtain the data.

Returns	Type	Description
pitch	float	The vertical angle of the view (degrees).

5.4.1.8 getRecordingLocation

Gets the recording location meta data of an opened image by viewer id. The recording location is returned as an object, having the following properties:

- id [string] : The recording location id, which is equal to the image id.
- recordedAt [string] : The recording date and time as an ISO 8601 DateTime string
- usedInMeasurements [boolean] : If this recording location may be used in measurements
- x [float] : The x-element of the recording location defined in the set srs name
- y [float] : The y-element of the recording location defined in the set srs name
- z (optional) [float] : The z-element of the recording location defined in the set srs name
- sx (optional) [float] : The x-element of the recording location standard deviations in meters
- sy (optional) [float] : The y-element of the recording location standard deviations in meters
- sz (optional) [float] : The z-element of the recording location standard deviations in meters
- height (optional) [float] : Used if the z-coordinate is unknown
- sHeight (optional) [float] : The standard deviation of the height variable

Parameter	Type	Description
viewerID	uint	The viewer id from which to obtain the data.

Returns	Type	Description
recordingLocation	Object	The recording location meta data as an object

5.4.1.9 getViewerBorderColor

Parameter	Type	Description
viewerID	uint	The viewer id from which to obtain the data.

Returns	Type	Description
color	uint	The color of the viewer window canvas as an RGB value. The two most significant bytes encode the color red.

5.4.1.10 getViewerClickMode

Gets the current viewer click mode.

Returns	Type	Description
clickMode	uint	The click mode

5.4.1.11 getViewerScreenshot

Gets a screenshot of a viewer. The screenshot is returned as an object, containing a `width`, `height`, and `data` property.

The `data` property consists of the base64 encoded pixel values of the image, where each pixel is represented by a uint. Each uint holds an RGB value of which the two most significant bytes reference the color red.

Parameter	Type	Description
viewerID	uint	The viewer of which to retrieve a screenshot.

Returns	Type	Description
screenshot	Object	Object containing screenshot data.

5.4.1.12 getYaw

Gets the horizontal orientation of a view in degrees.

Parameter	Type	Description
viewerID	uint	The viewer id of which to obtain the data from.

Returns	Type	Description
yaw	float	The horizontal angle of the view (degrees).

5.4.1.13 lookAtCoordinate

Centers the view direction of a viewer to a terrestrial coordinate as defined by the spatial reference system used within the API.

Parameter	Type	Description
viewerID	uint	The viewer of which to change its view direction.
x	float	The x-element of a terrestrial coordinate.
y	float	The y-element of a terrestrial coordinate.
z	float	The z-element of a terrestrial coordinate.

5.4.1.14 rotateDown

Rotate a viewer downwards by a certain angle in degrees.

Parameter	Type	Description

viewerID	uint	The viewer id of which to apply rotation on.
value	float	A rotation value in degrees.

5.4.1.15 rotateLeft

Rotate a viewer to the left by a certain angle in degrees.

Parameter	Type	Description
viewerID	uint	The viewer id of which to apply rotation on.
value	float	A rotation value in degrees.

5.4.1.16 rotateRight

Rotate a viewer to the right by a certain angle in degrees.

Parameter	Type	Description
viewerID	uint	The viewer id of which to apply rotation on.
value	float	A rotation value in degrees.

5.4.1.17 rotateUp

Rotate a viewer upwards by a certain angle in degrees.

Parameter	Type	Description
viewerID	uint	The viewer id of which to apply rotation on.
Value	float	A rotation value in degrees.

5.4.1.18 setBrightness

Sets the gamma of a viewer.

Parameter	Type	Description
viewerID	uint	The viewer id of which to apply rotation on.
brightness	float	A brightness value within the range of [0.1..3.0].

5.4.1.19 setHFov

Sets the horizontal field of view in degrees.

Parameter	Type	Description
viewerID	uint	The viewer id of which to apply the hFov to.

hFov	float	The horizontal field of view in degrees.

5.4.1.20 setOverlayAlpha

Sets the alpha value of the overlays on a specific viewer.

Parameter	Type	Description
viewerID	uint	The viewer id of which to apply the value to.
alpha	float	The alpha values for the overlays [0.0...1.0]

5.4.1.21 setOverlayDrawDistance

Sets the distance, in meters, used for drawing layers from the camera center within a cyclorama.

Parameter	Type	Description
viewerID	uint	The viewer id of which to apply the value to.
distance	float	The draw distance.

5.4.1.22 setPitch

Sets the vertical orientation of a view in degrees.

Parameter	Type	Description
viewerID	uint	The viewer id of which to apply the pitch to
pitch	float	The vertical angle of the view.

5.4.1.23 setViewerClickMode

Sets the viewer click mode. The click mode defines the behaviour of the viewer clicked event.The following modes are defined:

```
CLICK_MODE_X_Y = 1
```
The image coordinates within the current view. The center coordinate of the top left pixel is defined to be (0.5, 0.5). The viewer clicked event returns x and y values.

```
CLICK_MODE_H_V = 2
```
Horizontal and vertical spherical coordinates within the entire image (cyclorama). The center of the image is defined to be (0.0, 0.0). The viewer clicked event returns H and V values.

```
CLICK_MODE_VECTOR = 3
```

Unit vector x, y and z coordinates that define the location within the image. The center of the image is defined to be (0.0, 1.0, 0.0). The viewer clicked event returns x, y and z values.

Parameter	Type	Description
clickMode	uint	The click mode

5.4.1.24 setYaw

Sets the horizontal orientation of a view in degrees.

Parameter	Type	Description
viewerID	uint	The viewer id of which to apply the yaw to
yaw	float	The horizontal angle of the view (degrees).

5.4.1.25 showImageInformation

Opens the 'Image information' dialog containing the recording location metadata of the current image.

Parameter	Type	Description
viewerID	uint	The viewer ID.

5.4.1.26 showViewerLocationOnMap

Centers the map on the viewer and indicates its location highlighting it.

Parameter	Type	Description
viewerID	uint	An id that references a viewer.

5.4.1.27 showViewerPrintDialog

Shows the user a print-dialog for a viewer, in order to print the image of the specified viewer.

Parameter	Type	Description
viewerID	uint	The viewer of which the image will be printed.

5.4.1.28 showViewerSaveDialog

Shows the used a save-dialog for a viewer, in order to save the image of the specified viewer to disk.

Parameter	Type	Description
viewerID	uint	The viewer of which the image will be saved.

5.4.1.29 zoomViewerToMaxLevel

Sets the referenced viewer to its maximum zoom level.

Parameter	Type	Description
viewerID	uint	An id that references a viewer.

5.4.1.30 zoomViewerToMedLevel

Sets the referenced viewer to an intermediate zoom level.

Parameter	Type	Description
viewerID	uint	An id that references a viewer.

5.4.1.31 zoomViewerToMinLevel

Sets the referenced viewer to its minimum zoom level.

Parameter	Type	Description
viewerID	uint	An id that references a viewer.

5.4.1.32 clearMarker

Removes a marker from the drawing layer in a viewer.

Parameter	Type	Description
viewerID	uint	The ID of the viewer to remove the marker from.
drawingID	uint	The ID of the marker to be removed.

5.4.1.33 clearMarkers

Removes all markers from the drawing layer in a viewer.

Parameter	Type	Description
viewerID	uint	The ID of the viewer to remove the marker from.

5.4.1.34 drawMarkerAtHV

Draws a marker at position *H*,*V* in a viewer.

Parameter	Type	Description
viewerID	uint	ID of the viewer to draw the marker on
H	float	Horizontal direction in degrees
V	float	Vertical direction in degrees

label	String	Label for the marker (optional) Defaults to null.

Returns	Type	Description
drawingID	uint	ID of the added drawing

5.4.1.35 drawMarkerAtXY

Draws a marker at pixel x,y in a viewer (left, top = 0,0).

Parameter	Type	Description
viewerID	uint	ID of the viewer to draw the marker on
x	float	X coordinate of the pixel location
y	float	Y coordinate of the pixel location
label	String	Label for the marker (optional) Defaults to null

Returns	Type	Description
drawingID	uint	ID of the added drawing

5.4.1.36 drawMarkerInDirection

Draws a marker in direction x,y,z in a viewer.

Parameter	Type	Description
viewerID	uint	ID of the viewer to draw the marker on
x	float	X component of the direction
y	float	Y component of the direction
z	float	Z component of the direction
label	String	Label for the marker (optional). Defaults to null.

Returns	Type	Description
drawingID	uint	ID of the added drawing

5.4.1.37 setDrawingLayerVisible

Sets the visibility of the viewer drawing layer.

Parameter	Type	Description
Visible	Boolean	

5.4.1.38 setDrawingMode

Sets the drawing mode. The drawing mode determines the form of the drawing:

`DRAWING_MODE_POINT = 1`
Draws a point.

`DRAWING_MODE_URL_IMAGE = 2`
Draws an image. The image url must be set.

`DRAWING_MODE_CROSS_HAIR = 3`
Draws a cross-hair.

Parameter	Type	Description
mode	uint	Drawing mode

5.4.1.39 setMarkerColor

Sets the color used for drawing markers. This only applies to the non image-url drawing modes (dot, cross-hair, etc).

Parameter	Type	Description
color	uint	The color of a drawing as a 24-bit RGB value.

5.4.1.40 setMarkerImageURL

Sets the image url used for drawing.

Note that a crossdomain.xml should be available on the server where the image is hosted if not usingthe proxy parameter.

Parameter	Type	Description
imageURL	String	The URL to a small PNG or JPEG image.
useProxy	Boolean	Download images via the CycloMedia proxy service in case of a security sandbox violation.

5.4.1.41 setMarkerSize

Sets the size used for drawing markers.

Parameter	Type	Description
size	uint	Size of the drawing in pixels.

5.4.2 Events

5.4.2.1 hst_imageChanged

Indicates to the host that the image was changed within a certain viewer. This event occurs after the `openImage` function has been called, after clicking on a recording location, etc.

Parameter	Type	Description
viewerID	uint	The id of the viewer of which its image changed.

5.4.2.2 hst_imageCompleted

Indicates to the host that the entire high resolution image has been loaded.

Parameter	Type	Description
viewerID	uint	The id of the viewer of which its image changed.

5.4.2.3 hst_imageFailed

Indicates to the host that any part of the image could not be completely loaded. This means that the event is dispatched when either the preview or any segment fails to load.

Parameter	Type	Description
viewerID	uint	The id of the viewer of which its image changed.

5.4.2.4 hst_imagePreviewCompleted

Indicates to the host that the preview of an image completed. The preview is a low resolution image that is used to quickly show a result and give the user the experience of progress.

Parameter	Type	Description
viewerID	uint	The id of the viewer of which its image changed.

5.4.2.5 hst_imageSegmentLoaded

Indicates to the host that a segment of the high resolution image has been loaded.

Parameter	Type	Description
viewerID	uint	The id of the viewer of which its image changed.

5.4.2.6 hst_viewChanged

Indicates to the host that the current view is changed due to a change in zoom level and / or orientation.

Parameter	Type	Description
viewerID	uint	The id of the viewer of which its image changed

yaw	float	horizontal orientation of a view in degrees
pitch	float	vertical orientation of a view in degrees
hFov	float	horizontal field of view in degrees

5.4.2.7 *hst_viewClicked*

Indicates to the host that the user clicked on a viewer. The parameters of the function depend on the viewer click mode. See *setViewerClickMode* for more information.

Parameter	Type	Description
viewerID	uint	The id of the viewer of which its image changed
c0	float	Click parameter 0
c1	float	Click parameter 1
c2	float	Click parameter 2 (optional)

5.4.2.8 *hst_viewLoaded*

Indicates to the host that all image parts of the current view have been completely loaded.

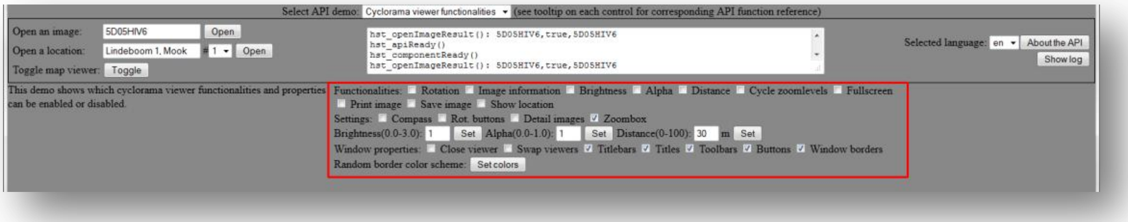
Parameter	Type	Description
viewerID	uint	The id of the viewer of which its image changed

5.4.2.9 *hst_markerClicked*

Indicates to the host that a marker from the drawing layer was clicked. Returns the position of the clicked marker similar to the "vector" viewer click mode.

Parameter	Type	Description
viewerID	uint	The id of the viewer of which its image changed
drawingID	uint	The id of the marker that was clicked
x	float	The x-element of the click direction vector
y	float	The y-element of the click direction vector
z	float	The z-element of the click direction vector

5.5 Cyclorama viewer functionalities


Select API demo: Cyclorama viewer functionalities (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

Toggle map viewer:

```
hst_openImageResult(): 5D05HIV6,true,5D05HIV6
hst_apiReady()
hst_componentReady()
hst_openImageResult(): 5D05HIV6,true,5D05HIV6
```

Selected language: en

This demo shows which cyclorama viewer functionalities and properties can be enabled or disabled.

Functionalities: □ Rotation □ Image information □ Brightness □ Alpha □ Distance □ Cycle zoomlevels □ Fullscreen □ Print image □ Save image □ Show location

Settings: □ Compass □ Rot. buttons □ Detail images ☑ Zoombox

Brightness(0.0-3.0): 1 Alpha(0.0-1.0): 1 Distance(0-100): 30 m

Window properties: □ Close viewer □ Swap viewers ☑ Titlebars ☑ Titles ☑ Toolbars ☑ Buttons ☑ Window borders

Random border color scheme:

The demo for this section can be found on the ‘Cyclorama viewer functionalities’ demo page. Corresponding example source code can be found in japi/demo-cyclorama-viewer-functionalities.html.

5.5.1 Functions

5.5.1.1 getCloseViewerEnabled

Returns true if the ‘close’ button is enabled for viewers.

Returns	Type	Description
enabled	Boolean	

5.5.1.2 getImageInformationEnabled

Returns true if the image information button is enabled for viewers.

Returns	Type	Description
enabled	Boolean	

5.5.1.3 getSwapViewersEnabled

Returns true if the ‘swap’ buttons are enabled for viewers.

Returns	Type	Description
enabled	Boolean	

5.5.1.4 getViewerBrightnessEnabled

Returns true if the brightness button is enabled for viewers.

Returns	Type	Description
enabled	Boolean	

5.5.1.5 getViewerCycleZoomLevelsEnabled

Returns true if the cycle zoom levels button button is enabled for viewers.

Returns	Type	Description

enabled	Boolean	

5.5.1.6 getViewerOverlayAlphaEnabled

Returns true if the vector data alpha slider button is enabled for viewers.

Returns	Type	Description
enabled	Boolean	

5.5.1.7 getViewerOverlayDrawDistanceEnabled

Returns true if the vector data distance slider button is enabled for viewers.

Returns	Type	Description
enabled	Boolean	

5.5.1.8 getViewerPrintImageEnabled

Returns true if the print image button is enabled for viewers.

Returns	Type	Description
enabled	Boolean	

5.5.1.9 getViewerSaveImageEnabled

Returns true if the save image button is enabled for viewers.

Returns	Type	Description
enabled	Boolean	

5.5.1.10 getViewerShowLocationEnabled

Returns true if the show location button is enabled for viewers.

Returns	Type	Description
enabled	Boolean	

5.5.1.11 setCloseViewerEnabled

Enables/disables the ‘close’ button for viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.12 setImageInformationEnabled

Enables/disables the image information button for viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.13 setSwapViewersEnabled

Enables/disables the ‘swap’ buttons for viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.14 setViewerBrightnessEnabled

Enables/disables the brightness button for viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.15 setViewerCycleZoomLevelsEnabled

Enables/disables zoom level button for viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.16 setViewerOverlayAlphaEnabled

Enables/disables vector data alpha slider for viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.17 setViewerOverlayDrawDistanceEnabled

Enables/disables vector data distance slider for viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.18 setViewerPrintImageEnabled

Enables/disables the print image button for viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.19 setViewerSaveImageEnabled

Enables/disables the save image button for cyclorama viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.20 setViewerShowLocationEnabled

Enables/disables the show location button for cyclorama viewers.

Parameter	Type	Description
enabled	Boolean	

5.5.1.21 getViewerBorderColorScheme

Returns the current color scheme used for drawing viewer borders.

Returns	Type	Description
colorScheme	uint[]	

5.5.1.22 getViewerCompassVisible

Returns true if the compass is visible in viewers.

Returns	Type	Description
visible	Boolean	

5.5.1.23 getViewerDetailImagesVisible

Returns true if high resolution images can be viewed inside panoramic images (when available).

Returns	Type	Description
visible	Boolean	Picture in picture visibility.

5.5.1.24 getViewerRotationEnabled

Returns true if rotation through keyboard, contextmenum toolbar and compass is allowed.

Returns	Type	Description
enabled	Boolean	

5.5.1.25 getViewerTitleVisible

Returns true if the title for viewers is visible.

Returns	Type	Description
visible	Boolean	

5.5.1.26 getViewerTitleBarVisible

Returns true if the titlebar for viewers is visible.

Returns	Type	Description
visible	Boolean	

5.5.1.27 getViewerToolBarButtonsVisible

Returns true if the button toolbar buttons for viewers are visible.

Returns	Type	Description
visible	Boolean	

5.5.1.28 getViewerToolBarVisible

Returns true if the button toolbar for viewers is visible.

Returns	Type	Description
visible	Boolean	

5.5.1.29 getViewerWindowBorderVisible

Returns true if the window border is visible for viewers.

Returns	Type	Description
visible	Boolean	

5.5.1.30 getViewerZoomBoxEnabled

Returns true if the zoom box (long press with mouse button in cyclorama to trigger zoom image in the left upper corner) is enabled.

Returns	Type	Description
enabled	Boolean	

5.5.1.31 setViewerBorderColorScheme

Sets the colorscheme to be used for viewers. Should be set before any viewers are opened.

Parameter	Type	Description
colorScheme	uint[]	

5.5.1.32 setViewerCompassVisible

Sets the visibility for the compass in viewers.

5.5.1.33 setViewerDetailImagesVisible

Sets the visibility for viewing high resolution images inside panoramic images (when available).

Parameter	Type	Description
visible	Boolean	Visibility of picture in picture.

5.5.1.34 setViewerRotationButtonsVisible

Sets the visibility of the rotation navigation buttons for viewers.

Parameter	Type	Description
visible	Boolean	

5.5.1.35 setViewerWindowBorderVisible

Sets the border visibility for viewers.

Parameter	Type	Description
visible	Boolean	

5.5.1.36 setViewerTitleVisible

Sets the visibility of the title for viewers.

Parameter	Type	Description
visible	Boolean	

5.5.1.37 setViewerTitleBarVisible

Sets the visibility of the titlebar for viewers.

Parameter	Type	Description
visible	Boolean	

5.5.1.38 setViewerToolBarVisible

Sets the visibility of the toolbar for viewers.

Parameter	Type	Description
visible	Boolean	

5.5.1.39 setViewerZoomBoxEnabled

Enables/disables the zoom box (long press with mouse button in a viewer to trigger zoom image in the left upper corner).

Parameter	Type	Description
enabled	Boolean	

5.5.2 Events

5.5.2.1 hst_showLocationRequested

Indicates to the host that the ‘Show location on map’-functionality has been triggered.

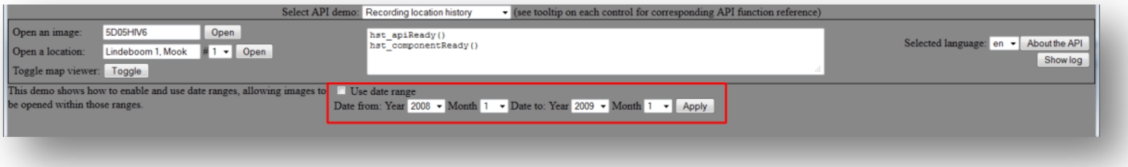
Parameter	Type	Description
viewerID	uint	The ID of the originating viewer
location	Object	An object containing x,y and z [Number] properties, indicating the position of the opened recording

5.5.2.2 hst_detailImagesVisibilityChanged

Indicates to the host that the visibility of the detail images has been changed

Parameter	Type	Description
visible	Boolean	The current visibility value

5.6 Recording location history



Select API demo: Recording location history (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

Toggle map viewer:

```
hst_apiReady()
hst_componentReady()
```

Selected language: en

This demo shows how to enable and use date ranges, allowing images to be opened within those ranges.

□ Use date range

Date from: Year 2008 Month 1 Date to: Year 2009 Month 1

The demo for this section can be found on the ‘Recording location history’ page. Corresponding example source code can be found in japi/demo-recording-dates.html.

5.6.1 Functions

5.6.1.1 getDateFrom

Returns the currently set dateFrom value.

Returns	Type	Description
date	String	An ISO 8601 DateTime string indicating all recordings newer than specified date

5.6.1.2 getDateTo

Returns the currently set dateTo value.

Returns	Type	Description
date	String	An ISO 8601 DateTime string indicating all recordings newer than specified date

5.6.1.3 getRecordingLocationColorFromDate

Returns the recording location color that is associated with the date parameter within the GlobeSpotter API.

Parameter	Type	Description
date	String	An ISO 8601 DateTime string indicating all recordings newer than specified date

Returns	Type	Description
color	uint	

5.6.1.4 getUseDateRange

If set to true, the recording locations layer will use the stored dateFrom and dateTo to get the desired recording locations.

Returns	Type	Description

visible	boolean	True if stored daterange should be used

5.6.1.5 setDateFrom

Sets the minimum date for requesting images within a certain date range or viewing recording locations between a certain date range. Its value has to be smaller than that of the `dateTo` and is not allowed to be `null`.

Parameter	Type	Description
dateFrom	String	The minimum date as an ISO 8601 DateTime string.

5.6.1.6 setDateTo

Sets the maximum date for requesting images within a certain date range or viewing recording locations between a certain date range. Its value has to be larger than that of the `dateFrom`. It is allowed to be <code>`null`</code> for requesting up until the current date time.

Parameter	Type	Description
dateTo	String	An ISO 8601 DateTime string indicating all recordings newer than specified date

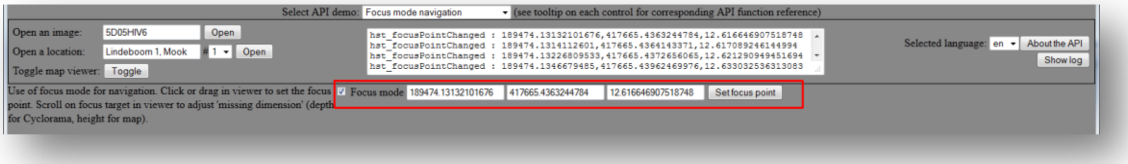
5.6.1.7 setUseDateRange

The useDateRange property influences how images are requested and which recording locations can be viewed within cycloramas and on the map. If set to true, images are only requested between a set date range and recording locations can only shown if their `recordedAt` property falls within the date range. If set to false, only the most recent recording locations are shown or used for requesting images.

The date range can be adjusted using the `setDateFrom` and `setDateTo` functions.

Parameter	Type	Description
useDateRange	Boolean	

5.7 Focus mode navigation


Select API demo: Focus mode navigation (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

```
hst_focusPointChanged : 189474.13132101676,417665.4363244784,12.616646907518748
hst_focusPointChanged : 189474.1314112601,417665.4364143371,12.617089246144994
hst_focusPointChanged : 189474.13226809533,417665.4372656065,12.621290949451694
hst_focusPointChanged : 189474.1346679485,417665.43962469976,12.633032536313083
```

Selected language: en

Use of focus mode for navigation. Click or drag in viewer to set the focus point. Scroll on focus target in viewer to adjust 'missing dimension' (depth for Cyclorama, height for map).

☑ Focus mode 189474.13132101676 417665.4363244784 12.616646907518748

The demo for this section can be found on the ‘Focus mode navigation’ demo page. Corresponding example source code can be found in japi/demo-focus-mode.html.

5.7.1 Functions

5.7.1.1 getFocusMode

Returns true if focus mode is enabled.

Returns	Type	Description
focusModeEnabled	Boolean	Returns true if focus mode is enabled

5.7.1.2 getFocusPoint

Gets the current position of the focus target used by the focus mode.

Returns	Type	Description
focusPoint	Object	An object having an *x*, *y* and optionally a *z* property. The z is only available if a viewer is opened.

5.7.1.3 setFocusPoint

Sets the current position of the focus target used by the focus mode.

Parameter	Type	Description
x	float	
y	float	
z	float	

5.7.1.4 setFocusMode

Enables or disables the focus mode. If enabled, a focus target will be visible and all viewers will center their view on that target.

Parameter	Type	Description
focusModeEnabled	Boolean	

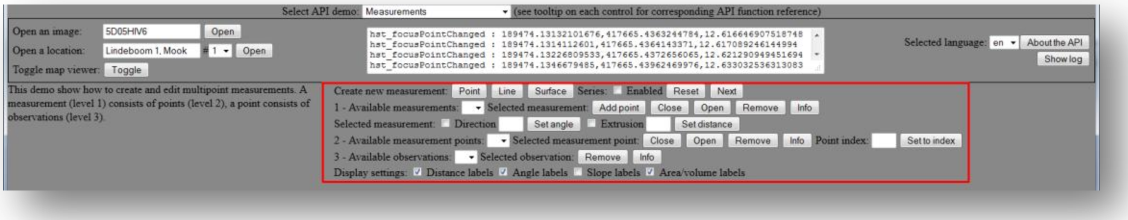
5.7.2 Events

5.7.2.1 hst_focusPointChanged

Indicates to the host that the focusPoint has been updated in FocusMode.

Parameter	Type	Description
x	float	
y	float	
z	float	

5.8 Measurements



Select API demo: Measurements (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

```
hst_focusPointChanged : 189474.13132101676,417665.4363244784,12.616646907518748
hst_focusPointChanged : 189474.1314112601,417665.4364143371,12.617089246144994
hst_focusPointChanged : 189474.13226809533,417665.4372656065,12.621290949451694
hst_focusPointChanged : 189474.1346679485,417665.43962469976,12.633032536313083
```

Selected language: en

This demo show how to create and edit multipoint measurements. A measurement (level 1) consists of points (level 2), a point consists of observations (level 3).

Series: ☐ Enabled

Selected measurement: ☐ Direction ☐ Extrusion

Display settings: ☑ Distance labels ☑ Angle labels ☐ Slope labels ☑ Area/volume labels

Open
Open
Toggle map viewer: Toggle
About the API
Show log
Create new measurement: Point Line Surface
Reset Next
1 - Available measurements: Selected measurement: Add point Close Open Remove Info
Set angle Set distance
2 - Available measurement points: Selected measurement point: Close Open Remove Info Point index: Set to index
3 - Available observations: Selected observation: Remove Info

The demo for this section can be found on the ‘Measurements’ demo page. Corresponding example source code can be found in japi/demo-measurements.html.

5.8.1 Functions

5.8.1.1 *addLineMeasurement*

Adds a line measurement entity to the list of entities, which can be used for measuring a 3D line. Series mode can be used to add points to the line measurement. To add points manually, see ‘addMeasurementPoint’. The function ‘closeMeasurement’ should be called when the measurement is finished.

Parameter	Type	Description
name	string	User defined name of the entity.

Returns	Type	Description
entityID	int	ID of the measurement.

5.8.1.2 *addMeasurementPoint*

Adds a point to an existing measurement (line of surface). If ‘focus mode’ is enabled, the point will be created on the current focus position using the opened images, otherwise an empty point is created to be completed using manual observations (click on images).

Parameter	Type	Description
entityID	int	ID of the measurement to add the point to.

Returns	Type	Description
pointID	int	ID of the point (-1 if adding failed)

5.8.1.3 *addPointMeasurement*

Adds a point measurement entity to the list of entities, which can be used for measuring a 3D point. 3D points need to be measured using two or more cyclorama windows. After calling this function the user is able to click in the cyclorama windows to get a 3D point. The function ‘closeMeasurement’ should be called when the measurement is finished.

Parameter	Type	Description
name	string	User defined name of the entity.

Returns	Type	Description
entityID	int	ID of the entity.

5.8.1.4 addSurfaceMeasurement

Adds a surface measurement entity to the list of entities, which can be used for measuring a 3D surface. Series mode can be used to add points to the surface measurement. To add points manually, see ‘addMeasurementPoint’. The function ‘closeMeasurement’ should be called when the measurement is finished.

Parameter	Type	Description
name	string	User defined name of the entity.

Returns	Type	Description
entityID	int	ID of the measurement.

5.8.1.5 cancelMeasurement

Cancels currently active measurement.

Parameter	Type	Description
entityID	uint	ID of the entity.

5.8.1.6 closeMeasurement

This function should be called when the desired measurement has been completed and the measurement can be finalized.

Parameter	Type	Description
entityID	int	ID of the entity.

Returns	Type	Description
Success	Boolean	If the measurement was valid and could be closed.

5.8.1.7 closeMeasurementPoint

This function should be called when the measurement point has been completed. New points can then be added to the measurement or the entire measurement can be closed.

Parameter	Type	Description
entityID	int	ID of the entity.
pointID	int	ID of the point to be closed.

Returns	Type	Description
success	Boolean	Can fail if the measurement was not opened for edit or when series mode is enabled.

5.8.1.8 createMeasurementPoint

Creates a point with a known position and adds it to an existing measurement.

Parameter	Type	Description
entityID	int	ID of the measurement to add the point to.
position	Object	A position described by an object having x,y and z properties [Number]

Returns	Type	Description
pointID	int	ID of the point.

5.8.1.9 getEntityData

Get the data object associated with an entity. The data of the object depends on the entity referenced.

Parameter	Type	Description
entityID	int	ID of the entity.

Returns	Type	Description
data	Object	The data object.

For regular measurement, the data object consists of the following properties:

Property	Type	Description
type	String	The type of the measurement: 'pointMeasurement' / 'lineMeasurement' / 'surfaceMeasurement' for regular measurements.
validEstimate	Boolean	Whether the measurement is valid or not.
pointArray	Array	A list of point data objects (see getMeasurementPointData)

area	float	The area of the measurement in m^2 (only for surface measurements)
areaStdDev	float	The standard deviation of the area estimation
totalDistance	float	The total length of a line measurement or the perimeter of a surface measurement
totalDistanceStdDev	float	The standard deviation of the totalDistance estimation
zDelta	float	The distance between the lowest and the hightest point of the measurement (only for line and surface measurements)
distances	Array	The distance per segment of the measurement (only for line and surface measurements)
angles	Array	The angles between two consecutive line segments (only for line and surface measurements)
slopes	Araay	The slope of each line segment (only for line and surface measurements)

5.8.1.10 getEntityDescription

Gets the description of a referenced entity.

Parameter	Type	Description
entityID	int	ID of the entity.

Returns	Type	Description
description	String	The description of the referenced entity.

5.8.1.11 getEntityName

Gets the name of a referenced entity.

Parameter	Type	Description
entityID	int	ID of the entity.

Returns	Type	Description
name	String	The name of the referenced entity.

5.8.1.12 getFocusEntity

Retrieves the unique id of the entity that has focus.

Returns	Type	Description
entityID	int	ID of the entity or -1 if no entity has focus.

5.8.1.13 getHideOverlaysWhenMeasuring

Returns true if overlays are hidden during measurements.

Returns	Type	Description
enabled	Boolean	

5.8.1.14 getMeasurementAnglesVisible

Returns true if angles are visible in the Cyclorama viewers for line and surface measurements.

Returns	Type	Description
visible	Boolean	

5.8.1.15 getMeasurementAreaVisible

Returns true if the area is visible in the Cyclorama viewers for surface measurements.

Returns	Type	Description
visible	Boolean	

5.8.1.16 getMeasurementDirection

Returns the direction that was added to the measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.

Returns	Type	Description
direction	float	The direction of the measurement [0.0...360.0]

5.8.1.17 getMeasurementDirectionEnabled

Returns true if a direction was added to the measurement.

Parameter	Type	Description

entityID	int	ID of the measurement.

Returns	Type	Description
enabled	Boolean	

5.8.1.18 getMeasurementDistancesVisible

Returns true if the distance is visible in the Cyclorama viewers for line and surface measurements.

Returns	Type	Description
visible	Boolean	

5.8.1.19 getMeasurementExtrusion

Returns the extrusion distance that was added to the measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.

Returns	Type	Description
distance	float	The extrusion distance of the measurement in metres.

5.8.1.20 getMeasurementExtrusionEnabled

Returns true if extrusion was added to the measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.

Returns	Type	Description
enabled	Boolean	

5.8.1.21 getMeasurementIDs

Returns the list of IDs of the current measurements.

Returns	Type	Description
list	Array	An array containing the IDs [int]

5.8.1.22 getMeasurementPointData

Gets the data object associated with a measurement point.

Parameter	Type	Description
entityID	int	ID of the measurement.
pointID	int	ID of the point.

Returns	Type	Description
data	Object	The data object

The data object consists of the following properties:

Property	Type	Description
pos	Object	An object with ‘x’, ‘y’ and ‘z’ properties containing the 3D position of the measurement point.
std	Object	An object with ‘x’, ‘y’ and ‘z’ properties containing the standard deviations of the 3D position of the measurement point in each dimension.
validEstimate	Boolean	Whether the measurement point is valid or not.
reliableEstimate	Boolean	Whether the measurement point is reliable or not.
observationArray	Array	A list of observation data objects (see getMeasurementPointObservationData)

5.8.1.23 getMeasurementPointIDs

Returns the list of IDs of the measurement points in the indicated measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.

Returns	Type	Description
list	Array	An array containing the IDs [int]

5.8.1.24 getMeasurementPointIndex

Gets the index of a measurement point within the measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.
pointID	int	ID of the point.

Returns	Type	Description
index	int	

5.8.1.25 getMeasurementPointObservationData

Gets the data object associated with an observation within a measurement point.

Parameter	Type	Description
entityID	int	ID of the measurement.
pointID	int	ID of the point.
imageID	String	ID of the image in which the observation was done.

Returns	Type	Description
data	Object	

The data object consists of the following properties:

Property	Type	Description
direction	Object	An object with ‘x’, ‘y’ and ‘z’ properties containing the direction in which the observation was clicked.
origin	Object	An object with ‘x’, ‘y’ and ‘z’ properties containing the origin from which the observation was clicked.
imageID	String	The ID of the image in which the observation was performed.

5.8.1.26 getMeasurementPointObservationImageIDs

Returns the list of IDs of the images in which the observations for the indicated measurement point where done.

Parameter	Type	Description
entityID	int	ID of the measurement.
pointID	int	ID of the point.

Returns	Type	Description
list	Array	A list of imageIDs [String]

5.8.1.27 getMeasurementSeriesModeEnabled

Returns true if line and surface measuring is currently in series mode (points are automatically created on consecutive clicks).

Returns	Type	Description

enabled	Boolean	

5.8.1.28 getMeasurementSlopesVisible

Returns true if slopes are visible in the Cyclorama viewers for line and surface measurements.

Returns	Type	Description
visible	Boolean	

5.8.1.29 getMeasurementSmartClickModeEnabled

Returns true if Smart Click measuring is enabled. Adding an observation to an empty measurement will then result in a request to the Smart Click Measurements (SCM) service.

Returns	Type	Description
visible	Boolean	

5.8.1.30 openMeasurement

Opens an existing measurement. Measurement points can be added,removed or updated.

Parameter	Type	Description
entityID	int	ID of measurement entity.

5.8.1.31 openMeasurementPoint

Opens an existing measurement poit. All observations and helper lines are visible again. The user can modify observations and store the changed measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.
pointID	int	ID of the point.

Returns	Type	Description
success	Boolean	Can fail if the measurement was not opened for edit or when series mode is enabled.

5.8.1.32 removeAllEntities

Remove all entities from the application.

5.8.1.33 removeEntity

Remove an entity so it is no longer drawn by the application.

Parameter	Type	Description
entityID	int	A reference to an entity.

5.8.1.34 removeMeasurementPoint

Remove the point from the indicated measurement.

Parameter	Type	Description
entityID	int	ID of the measurement
pointID	int	ID of the point.

Returns	Type	Description
success	Boolean	Can fail if the measurement was not opened for edit or when series mode is enabled.

5.8.1.35 removeMeasurementPointObservation

Remove the observation from the indicated measurement point.

Parameter	Type	Description
entityID	int	ID of the measurement
pointID	int	ID of the point.
imageID	String	ID of the image in which the observation was done.

Returns	Type	Description
success	Boolean	Can fail if the measurement was not opened for edit or when series mode is enabled.

5.8.1.36 setEntityDescription

Sets the description of a referenced entity.

Parameter	Type	Description
entityID	int	A reference to an entity.
description	string	A description of the entity or something referenced by the entity.

5.8.1.37 setFocusEntity

Sets the focus-entity by ID.

Parameter	Type	Description

entityID	int	A reference to an entity.

5.8.1.38 setHideOverlaysWhenMeasuring

Enables/disables hiding of overlays when performing measurements.

Parameter	Type	Description
enabled	Boolean	Set to true if overlays should be hidden when making measurements.

5.8.1.39 setMeasurementAnglesVisible

Sets the visibility of the angles in the Cyclorama viewers for line and surface measurements.

Parameter	Type	Description
visible	Boolean	

5.8.1.40 setMeasurementAreaVisible

Sets the visibility of the area in the Cyclorama viewers for surface measurements.

Parameter	Type	Description
visible	Boolean	

5.8.1.41 setMeasurementDirection

Adds a direction to the measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.
direction	float	The direction of the measurement [0.0...360.0]

5.8.1.42 setMeasurementDirectionEnabled

Enables/disables the direction of the measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.
enabled	Boolean	

5.8.1.43 *setMeasurementDistancesVisible*

Sets the visibility of the distance in the Cyclorama viewers for line and surface measurements.

Parameter	Type	Description
visible	Boolean	

5.8.1.44 *setMeasurementExtrusion*

Sets an extrusion distance for the measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.
distance	float	The extrusion distance of the measurement in metres.

5.8.1.45 *setMeasurementExtrusionEnabled*

Adds extrusion to the measurement.

Parameter	Type	Description
entityID	int	ID of the measurement.
enabled	Boolean	

5.8.1.46 *setMeasurementSeriesModeEnabled*

Enables/disables series mode for line and surface measuring (when enabled points are automatically created on consecutive clicks).

Parameter	Type	Description
enabled	Boolean	

5.8.1.47 *setMeasurementSlopesVisible*

Sets the visibility of slopes in the Cyclorama viewers for line and surface measurements.

Parameter	Type	Description
visible	Boolean	

5.8.1.48 *setMeasurementSmartClickModeEnabled*

Enables/disables Smart Click measuring. In Smart Click mode adding an observation to an empty measurement will result in a request to the Smart Click Measurements (SCM) service.

Parameter	Type	Description
enabled	Boolean	

5.8.2 Events

5.8.2.1 hst_entityDataChanged

Indicates to the host that the data of an entity is updated.

Parameter	Type	Description
entityID	int	The id of the entity of which its data is updated.
entityData	Object	The data of the entity that updated of which its contents depend on its type.

5.8.2.2 hst_entityFocusChanged

Indicates to the host that the focus of an entity has changed.

Parameter	Type	Description
entityID	int	The id of the entity or -1 if the focus changed to null.

5.8.2.3 hst_measurementCanceled

Indicates to the host that the user canceled the last active measurement. A measurement is canceled when the cancelMeasurement is called or if it was removed using `removeEntity` function.

Parameter	Type	Description
entityID	int	

5.8.2.4 hst_measurementClosed

Indicates to the host that the created measurement has been finalized. After a measurement is finalized it is locked from edition. Measurements are considered entities and can be deleted using the `removeEntity` function.

Parameter	Type	Description
entityID	int	
entityData	Object	Measurement data of which the contents depends on the entity type

5.8.2.5 *hst_measurementCreated*

Indicates to the host that a new measurement has been created. A measurement can only be edited until it is considered finished (after calling `closeMeasurement`) after which it is locked. After creating a new measurement, it can be interacted with using so called observations. An observation is a 2D point in an image that represents the projection the 3D point that is to be measured. Observations are added by clicking on an image, and can be manipulated by clicking on another location in the same image, or by dragging it around using mouse interaction. A measurement can only be created and edited through user interaction. After a measurement is finalized, it is locked from edition. Measurements are considered entities and can be deleted using the `removeEntity` function.

Parameter	Type	Description
entityID	int	
entityType	String	The type of measurement created

5.8.2.6 *hst_measurementModeChanged*

Indicates to the host that the Cyclorama viewers are (no longer) in measurement mode.

Parameter	Type	Description
enabled	Boolean	Indicates the current state of the measurement mode.

5.8.2.7 *hst_measurementOpened*

Indicates to the host that an existing measurement was opened again for edition.

Currently supported types:

```
areaEntityType
heightEntityType
volumeEntityType
pointEntityType
lineEntityType
```

Parameter	Type	Description
entityID	int	
entityType	String	The type of measurement opened.

5.8.2.8 *hst_measurementPointAdded*

Indicates to the host that the measurement point was added to the indicated measurement.

Parameter	Type	Description

entityID	int	
pointID	int	

5.8.2.9 hst_measurementPointClosed

Indicates to the host that the measurement point was closed on the indicated measurement.

Parameter	Type	Description
entityID	int	
pointID	int	

5.8.2.10 hst_measurementPointObservationAdded

Indicates to the host that an observation was added to the indicated measurement point.

Parameter	Type	Description
entityID	int	
pointID	int	
imageID	String	The ID of the image in which the observation was done

5.8.2.11 hst_measurementPointObservationRemoved

Indicates to the host that an observation was removed from the indicated measurement point.

Parameter	Type	Description
entityID	int	
pointID	int	
imageID	String	The ID of the image in which the observation was done

5.8.2.12 hst_measurementPointObservationUpdated

Indicates to the host that an observation was updated on the indicated measurement point.

Parameter	Type	Description
entityID	int	
pointID	int	
imageID	String	The ID of the image in which the observation was done

5.8.2.13 hst_measurementPointOpened

Indicates to the host that the measurement point was opened on the indicated measurement.

Parameter	Type	Description
entityID	int	
pointID	int	

5.8.2.14 hst_measurementPointRemoved

Indicates to the host that the measurement point was removed from the indicated measurement.

Parameter	Type	Description
entityID	int	
pointID	int	

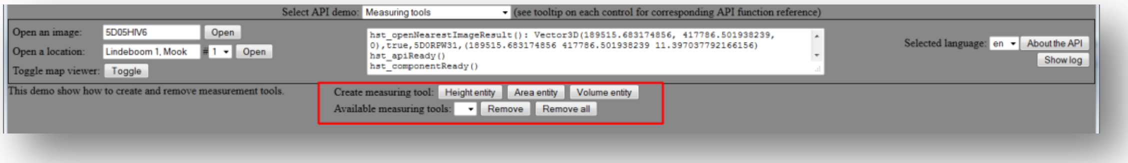
5.8.2.15 hst_measurementPointUpdated

Indicates to the host that the measurement point was updated on the indicated measurement.

Parameter	Type	Description
entityID	int	
pointID	int	

5.9 Measuring tools

5.9.1 Functions



Select API demo: Measuring tools (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

```
hst_openNearestImageResult(): Vector3D(189515.683174856, 417786.501938239,
0),true,5D0RPW31,(189515.683174856 417786.501938239 11.397037792166156)
hst_apiReady()
hst_componentReady()
```

Selected language: en

This demo show how to create and remove measurement tools.

The demo for this section can be found on the ‘Measuring tools’ demo page. Corresponding example source code can be found in japi/demo-measuring-tools.html.

The measuring tools also use the following functions/callbacks from the previous chapter:

- getEntityData
- getEntityDescription
- getFocusEntity
- setEntityDescription
- setFocusEntity
- hst_entityDataChanged
- hst_entityFocusChanged

5.9.1.1 addAreaEntity

Adds a height entity to the list of entities, which can be used for measuring the height and/or position of an object.

Parameter	Type	Description
name	String	User defined name of the entity.
height	float	The initial height.
point1	Object	A 2D position referenced in the coordinate system as used by the API.
point2	Object	A 2D position referenced in the coordinate system as used by the API.
z	float	The height is measured relative to this value.

Returns	Type	Description
entityID	uint	ID of the entity.

5.9.1.2 addHeightEntity

Adds a height entity to the list of entities, which can be used for measuring the height and/or position of an object.

Parameter	Type	Description
name	string	User defined name of the entity.
height	float	The initial height.
point	object	A 3D position referenced in the coordinate system as used by the API. Object height is measured relative to this position.

Returns	Type	Description
entityID	uint	ID of the entity.

5.9.1.3 addVolumeEntity

Adds a volume entity to the list of entities, which can be used for measuring the height, ground area, volume and/or perimeter of an object.

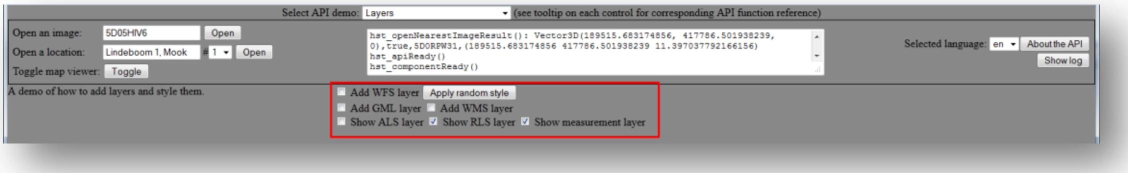
Parameter	Type	Description
name	string	User defined name of the entity.
height	float	The initial height.
points	object[]	A list of 2D positions referenced in the coordinate system as

		used by the API.
z	float	The height is measured relative to this value.

Returns	Type	Description
entityID	uint	ID of the entity.

5.10 Layers

5.10.1 Functions



Select API demo: Layers (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

Toggle map viewer:

```
hst_openNearestImageResult(): Vector3D(189515.683174856, 417786.501938239,
0),true,5D0RPW31,(189515.683174856 417786.501938239 11.397037792166156)
hst_apiReady()
hst_componentReady()
```

Selected language: en

A demo of how to add layers and style them.

□ Add WFS layer

□ Add GML layer □ Add WMS layer

□ Show ALS layer ☑ Show RLS layer ☑ Show measurement layer

The demo for this section can be found on the 'Layers' demo page. Corresponding example source code can be found in japi/demo-layers.html.

5.10.1.1 *addGMLLayer*

Use GML to add a new vector layer onto all viewers and the map.

Parameter	Type	Description
name	String	Name of the layer to be added.
gml	String	The GML data.
srsName	String	The SRS name of the GML data.
style	uint or String	Either a color or styled layer descriptor (Optional). The color is a 24-bits RGB value of which the two most significant bytes specify the color red, and the SLD/SE is of version 1.1. The default value is 0xFF.
showInViewer	Boolean	Default is true.
showInMap	Boolean	Default is true.
minZoomLevel	int	The zoom level from which to view the layer on the map (Optional). The default is -1, meaning that the featureZoomLevel is used. The feature zoom level is used as the minimum zoom level for recording and address location features.

Returns	Type	Description
layerID	uint	A reference to the added layer.

5.10.1.2 *addWFSLayer*

Add a new WFS layer to all viewers and the map.

Due to security restrictions in the flash player, it is sometimes required to route data via the CycloMeda proxy server. This restriction occurs when the domain on which the WFS service is host does not explicitly allow a flash client from another domain to use its data (no crossdomain.xml available).

Input parameter	Type	Description
name	string	Name of the layer
url	string	URL of the WFS service used to draw feature data
typeName	string	Name(s) of the features to be drawn on the layer
version	string	The WFS version to be used. For example `"1.1.0"`
style	uint or String	Either a color or styled layer descriptor (Optional). The color is a 24-bits RGB value of which the two most significant bytes specify the color red, and the SLD/SE is of version 1.1. The default value is 0xFF.
showInViewer	Boolean	Default is true.
showInMap	Boolean	Default is true.
minZoomLevel	int	The zoom level from which to view the layer on the map (Optional). The default is -1, meaning that the featureZoomLevel is used. The feature zoom level is used as the minimum zoom level for recording and address location features.
useProxy	boolean	Route WFS data through a CycloMedia proxy server (optional). Default is false

Returns	Type	Description
layerID	uint	A reference to the added layer.

5.10.1.3 addWMSLayer

Add a new WMS layer to the map. Due to security restrictions in the flash player, it is sometimes required to route data via the CycloMeda proxy server. This restriction occurs when the domain on which the WFS service is host does not explicitly allow a flash-client from another domain to use its data.

Parameter	Type	Description
name	String	Name of the layer
url	String	URL of the WMS service used to draw feature data
layer	String	Layer name on the server
version	String	The WMS version to be used

transparent	Boolean	Whether or not to display transparent pixels (if false, otherwise transparent pixels will be white)
tiled	Boolean	Whether or not the layer is tiled.
maxZoomLevel	uint	Specifies until which zoomlevel this layer should be visualized. Zoomlevel 0 is most top. Zoomlevel > 0 is zoomed in on map
minZoomLevel	uint	Specifies at which zoomlevel this layer should be visualized. Zoomlevel 0 is most top. Zoomlevel > 0 is zoomed in on map. Default is 0
bgcolor	uint	The background color of the layer if it is not transparent (optional). Default is 0xFFFFFF.
useProxy	Boolean	Route WMS data through a CycloMedia proxy server (optional). Default is false.
getFeatureInfo	Boolean	Specifies if the WMS feature info panel should be visible when a users clicks on a WMS feature. Default is false

Returns	Type	Description
layerID	uint	A reference to the added layer.

5.10.1.4 applyStyle

Apply a style (SLD or plain color) to a vector layer.

Parameter	Type	Description
layerID	uint	A reference to the layer to be removed
style	Object	Object that can be either a `uint` to specify a 24-bits RGB value or a `String` specifying a styled layer descriptor SLD/SE 1.1 compatible layer description in XML format

5.10.1.5 getAddressLocationsVisible

Gets the visibility of the address locations as a layer.

Returns	Type	Description
visible	Boolean	Layer visibility.

5.10.1.6 getMeasureLayerVisible

Returns true if the measure layer is visible.

Returns	Type	Description
visible	Boolean	True if measure layer is visible

5.10.1.7 getRecordingLocationsVisible

Gets the visibility of the recording locations as layer.

Returns	Type	Description
visible	Boolean	Layer visibility.

5.10.1.8 removeLayer

Remove a layer.

Parameter	Type	Description
layerID	uint	A reference to the layer to be removed

5.10.1.9 setAddressLocationsVisible

Sets the visibility of the address locations layer.

Parameter	Type	Description
visible	Boolean	

5.10.1.10 setBaseLayer

Sets the base layer for the map. All layers added after the map is opened are bound by the properties of the base layer.

A base layer can have any of the following types:

1) WMS - A raster data based layer.
2) WFS – A vector based layer.
3) OSM – OpenStreetMap raster/tile based layer

The properties of the base layer are set using the **params** parameter, the value of which is an object which shoud have the following properties:

- Common properties:
    - **url** [String] – the URL from which to obtain the layer data.
    - **maxExtent** [Object] – the maximum extent of the map, defined as an object having *xmin*, *ymin*, *xmax*, *ymax* [Number] as properties.
    - **maxResolution** [Number] – The maximum resolution of the map. It is defined as the maximum amount of distance in units per pixel of the rendered map.
    - **maxZoomLevel** [uint] – The maximum zoom level of the map. This value defines the total amount of zoom levels as *maxZoomLevel* + 1.
    - **featureZoomLevel** [uint] – The level above which common features will be visible.
- WMS properties

    - **layers** [String] – A string of comma separated layer name. E.g., *“parcels,streets”*
    - **version** [String] **–** The WMS version to be used.
    - **styles** (optional) [String] **–** Default uses the default styles. A string of comma separated style names. E.g., *“parcels_style, streets_style”*. The list must have the same length of comma separated values as the **layers** property.
    - **transparent** (optional) [Boolean] **–** Default is false. If true, the background color is set to transparent.
    - **tiled** (optional) [Boolean] - Default is false. If true, the WMS layer is obtained using tiles.
    - **bgcolor** (optional) [uint] – Default is 0xFFFFFF. The background color of the WMS layer in RGB format. The two must significant bytes encode the color red.
- WFS properties:
    - **typeName** [String] – A string of comma separated feature names. E.g. *“bridges,trees”*.
    - **version** [String] – The WFS version to be used.
    - **color** (optional) [uint] **–** Default is 0x0000FF. The color of the features drawin on the layer in RGB format. The two most significant bytes encode the color red, meaning the default color is blue.

There are no OSM specific properties when using the **params** object.

Parameter	Type	Description
name	String	Name of the base layer
type	uint	Type of the base layer
srsName	String	The Spatial Reference System to be used for the map
params	Object	Layer parameters specific to the layer type
useProxy	Boolean	Route layer data through a CycloMedia proxy server (optional). Default is false.

5.10.1.11 setMeasureLayerVisible

Sets the visibility of the measure layer.

Parameter	Type	Description
visible	Boolean	

5.10.1.12 setRecordingLocationsVisible

Sets the visibility of the recording locations layer.

Parameter	Type	Description
visible	Boolean	

5.10.2 Event

5.10.2.1 hst_featureClicked

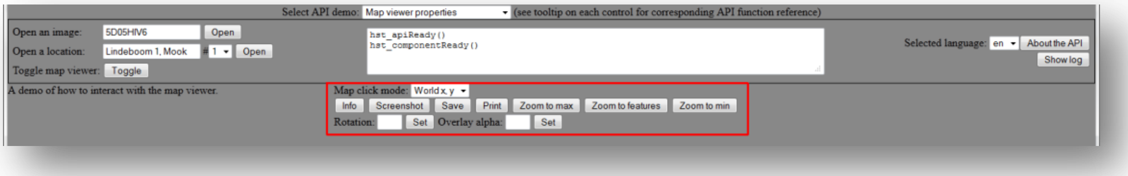
Indicates to the host that a feature has been clicked. This callback returns the contents of the feature as an object.

It is assumed that the features are simple features, meaning that it has a single geometry property and other than that no complex XML elements.

Parameter	Type	Description
featureData	Object	The data of a feature.

5.11 Map properties

5.11.1 Functions



Select API demo: Map viewer properties (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6

Open a location: Lindeboom 1, Mook # 1

```
hst_apiReady()
hst_componentReady()
```

Selected language: en

A demo of how to interact with the map viewer.

Map click mode: World x, y

The demo for this section can be found on the ‘Map viewer properties’ demo page. Corresponding example source code can be found in japi/demo-map-viewer-properties.html.

(* = Only available in Oblique-version of API)

5.11.1.1 *getMapCenter*

Returns the center of the map.

Returns	Type	Description
center	Object	An object having *x* and *y* as properties.

5.11.1.2 *getMapClickMode*

Gets the click-mode of the map.

Returns	Type	Description
clickMode	uint	The click mode.

5.11.1.3 *getMapExtent*

Retrieves the maximum bounds of the map.

Returns	Type	Description
extent	Object	An object having *xmin*, *ymin*, *xmax* and *ymax* as properties.

5.11.1.4 *getMapOverlayAlpha*

Returns the current alpha value used for overlays.

Returns	Type	Description
alpha	float	The current alpha value [0.0...1.0]

5.11.1.5 getMapRotation

Gets the rotation angle of the map.

Returns	Type	Description
angle	Number	The current rotation in degrees.

5.11.1.6 getMapScreenshot

Gets a screenshot of the map. The screenshot is returned as an object, containing a width, height and data property. The data property consists of the base64 encoded pixel values of the image, where each pixel is represented by a uint. Each uint holds an RGB value of which the two most significant bytes reference the color red.

Returns	Type	Description
screenshot	Object	An object containing screenshot data.

5.11.1.7 getObliqueModeEnabled*

Returns true if ‘oblique mode’ is enabled for the map viewer.

Returns	Type	Description
enabled	Boolean	

5.11.1.8 getMapZoom

Returns the zoom level of the map.

Returns	Type	Description
zoom	Number	The zoom level of the map.

5.11.1.9 obliqueViewFromBehind*

Rotates the view 180 degrees. Equivalent to choosing the ‘view from’ oblique recording.

5.11.1.10 obliqueViewFromLeft*

Rotates the view 90 degrees counter clockwise. Equivalent to choosing the ‘view from left side’ oblique recording.

5.11.1.11 obliqueViewFromRight*

Rotates the view 90 degreesclockwise. Equivalent to choosing the ‘view from right side’ oblique recording.

5.11.1.12 setMapCenter

Sets the centre of the map. Either with a point-object, or with two loose coordinates.

Parameter	Type	Description
center	Object	Object with 'x', 'y' (Note: Either specify center as object OR x,y as numbers)
x	Number	x-Coord (Note: Either specify center as object OR x,y as numbers)
y	Number	y-Coord (Note: Either specify center as object OR x,y as numbers)

5.11.1.13 setMapClickMode

Retrieves the click-mode of the map:

`WINDOW = 0`

Returns window-coordinates to the host if the user clicks on the map.

`WORLD = 1`

Returns world-coordinates to the host if the user clicks on the map.

Parameter	Type	Description
clickMode	uint	The click mode.

5.11.1.14 setMapExtent

Sets the bounds of the map.

Parameter	Type	Description
bounds	Object	Object with 'xmin’, 'ymin', 'xmax', ‘ymax'

5.11.1.15 setMapOverlayAlpha

Sets the current alpha value used for overlays.

Parameter	Type	Description
alpha	float	The current alpha value [0.0...1.0]

5.11.1.16 setMapRotation

Sets the rotation angle of the map.

Parameter	Type	Description
angle	Number	The rotation angle to be set in degrees.

5.11.1.17 setMapZoom

Sets the zoom level of the map.

Parameter	Type	Description
zoom	Number	The zoom level of the map

*5.11.1.18 setObliqueModeEnabled**

Enables/disables ‘oblique mode’ for the map viewer.

Parameter	Type	Description
enabled	Boolean	

5.11.1.19 showMapPrintDialog

Shows the user a print dialog for the map, in order to print the image of the map.

5.11.1.20 showMapSaveDialog

Shows the user a save dialog for the map, in order to save the image of the map to disk.

5.11.1.21 zoomMapToFeatureLevel

Zooms the map to the level where common features are visible.

5.11.1.22 zoomMapToMaxLevel

Zooms the map to its maximum level.

5.11.1.23 zoomMapToMinLevel

Zooms the map to its minimum level.

5.11.2 Events

5.11.2.1 hst_mapClicked

Indicates to the host that the map was clicked.

Parameter	Type	Description
x	float	x-Coord of click in either world or window coords
y	float	y-Coord of click in either world or window coords

5.11.2.2 hst_mapExtentChanged

Indicates to the host that the map extent has changed. Note that adapting the zoom or centre of the map changes the extent too.

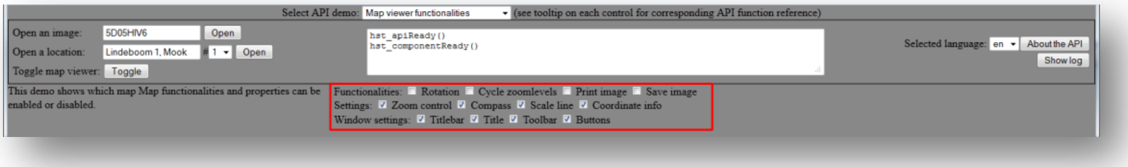
Parameter	Type	Description
extent	Object	Bounds of the map (top, bottom, left, right)
center	Object	Position of the centre of the map (x, y)
zoom	float	Zoom of the map

5.11.2.3 hst_mapInitialized

Indicates to the host that the map is ready for navigation/manipulation etc.

5.12 Map functionality

5.12.1 Functions



Select API demo: Map viewer functionalities (see tooltip on each control for corresponding API function reference)

Open an image: 5D05HIV6 Open

Open a location: Lindeboom 1, Mook # 1 Open

Toggle map viewer: Toggle

```
hst_apiReady()
hst_componentReady()
```

Selected language: en

This demo shows which map Map functionalities and properties can be enabled or disabled.

Functionalities: □ Rotation □ Cycle zoomlevels □ Print image □ Save image

Settings: ☑ Zoom control ☑ Compass ☑ Scale line ☑ Coordinate info

Window settings: ☑ Titlebar ☑ Title ☑ Toolbar ☑ Buttons

The demo for this section can be found on the ‘Map viewer functionality’ demo page. Corresponding example source code can be found in japi/demo-map-viewer-functionality.html.

5.12.1.1 getMapCompassVisible

Returns true if the compass is visible on the map.

Returns	Type	Description
visible	Boolean	

5.12.1.2 getMapCoordinateInfoVisible

Returns true if the coordinate info for the mouse position is visible on the map.

Returns	Type	Description
visible	Boolean	

5.12.1.3 getMapCycleZoomLevelsEnabled

Returns true if cycle zoom levels is enabled for the map viewer.

Returns	Type	Description
enabled	Boolean	

5.12.1.4 getMapObliqueDirectionsEnabled

Returns true if the view direction buttons are enabled for the map viewer in oblique mode.

Returns	Type	Description
enabled	Boolean	

5.12.1.5 getMapObliqueToggleVisible

Returns true if the toggle button to enable oblique mode is visible for the map viewer.

Returns	Type	Description
enabled	Boolean	

5.12.1.6 getMapPrintImageEnabled

Returns true if print image is enabled for the map viewer.

Returns	Type	Description
enabled	Boolean	

5.12.1.7 getMapRotationEnabled

Returns true if rotation is enabled through toolbar buttons, compass or keyboard shortcuts for the map viewer.

Returns	Type	Description
enabled	Boolean	

5.12.1.8 getMapSaveImageEnabled

Returns true if save image is enabled for the map viewer.

Returns	Type	Description
enabled	Boolean	

5.12.1.9 getMapScaleLineVisible

Returns true if the scale line is visible on the map.

Returns	Type	Description
visible	Boolean	

5.12.1.10 getMapTitleBarVisible

Returns true if the title bar is visible for the map viewer.

Returns	Type	Description
visible	Boolean	

5.12.1.11 getMapTitleVisible

Returns true if the title is visible for the map viewer.

Returns	Type	Description
visible	Boolean	

5.12.1.12 getMapToolBarButtonsVisible

Returns true if the tool bar buttons are visible for the map viewer.

Returns	Type	Description
visible	Boolean	

5.12.1.13 getMapToolBarVisible

Returns true if the tool bar is visible for the map viewer.

Returns	Type	Description
visible	Boolean	

5.12.1.14 getMapWindowBorderVisible

Returns true if the window border is visible for the map viewer.

Returns	Type	Description
visible	Boolean	

5.12.1.15 getMapZoomControlVisible

Returns true if the zoom control is visible on the map.

Returns	Type	Description
visible	Boolean	

5.12.1.16 setMapCompassVisible

Sets the visibility of the compass on the map.

Parameter	Type	Description
visible	Boolean	

5.12.1.17 setMapCoordinateInfoVisible

Sets the coordinate info for the mouse position is visible on the map.

Parameter	Type	Description
visible	Boolean	

5.12.1.18 setMapCycleZoomLevelsEnabled

Enables/disables cycle zoom levels in the map viewer.

Parameter	Type	Description
enabled	Boolean	

5.12.1.19 setMapObliqueDirectionsEnabled

Enables/disables the view direction buttons for the map viewer in oblique mode.

Parameter	Type	Description
enabled	Boolean	

5.12.1.20 setMapObliqueToggleVisible

Sets the visibility of the toggle button to enable oblique mode for the map viewer.

Parameter	Type	Description
enabled	Boolean	

5.12.1.21 setMapPrintImageEnabled

Enables/disables print image for the map viewer.

Parameter	Type	Description
enabled	Boolean	

5.12.1.22 setMapRotationEnabled

Enables/disables rotation through toolbar buttons, compass or keyboard shortcuts for the map viewer.

Parameter	Type	Description
enabled	Boolean	

5.12.1.23 setMapSaveImageEnabled

Enables/disables save image for the map viewer.

Parameter	Type	Description
enabled	Boolean	

5.12.1.24 setMapScaleLineVisible

Sets the visibility of the scale line on the map.

Parameter	Type	Description
visible	Boolean	

5.12.1.25 setMapTitleBarVisible

Sets the visibility of the title bar for the map viewer.

Parameter	Type	Description
visible	Boolean	

5.12.1.26 setMapTitleVisible

Sets the visibility of the title for the map viewer.

Parameter	Type	Description
visible	Boolean	

5.12.1.27 setMapToolBarButtonsVisible

Sets the visibility of the tool bar buttons for the map viewer.

Parameter	Type	Description
visible	Boolean	

5.12.1.28 setMapToolBarVisible

Sets the visibility of the tool bar for the map viewer.

Parameter	Type	Description
visible	Boolean	

5.12.1.29 setMapWindowBorderVisible

Sets the visibility of the window border for the map viewer.

Parameter	Type	Description

visible	Boolean	

5.12.1.30 setMapZoomControlVisible

Sets the visibility of the zoom control on the map.

Parameter	Type	Description
visible	Boolean	

6 Useful links

Flash player	Flash debug player	http://www.adobe.com/support/flashplayer/downloads.html
Documentation		http://www.cyclomedia.nl/service-en-support/service-en-support-voor-ontwikkelaars/
Demo WFS	Demo vector data	http://www.globedata.nl/demo/wfs
Demo WMS	Demo raster data	http://www.globedata.nl/demo/wms
Notepad++	Simple editor with syntax highlighting	http://notepad-plus-plus.org/
Fiddler	Web debugger	http://www.fiddler2.com/fiddler2/
Security	Security sandboxes	http://help.adobe.com/en_US/ActionScript/3.0_ProgrammingAS3/WS5b3ccc516d4fbf351e63e3d118a9b90204-7e3f.html
Cross domain	Example remote security file	https://www.globespotter.nl/crossdomain.xml
Firebug	Mozilla firefox add-on (debugger)	Download via Firefox
MSE	Microsoft Script Editor (debugger)	Installed with Office: C:\Program Files\Common Files\Microsoft Shared\OFFICE12\ http://www.hlrnet.com/vbweb/leren_mse.htm