SMES Web Services

Application Programming Interface

Version 2.09.02

Last Updated 20/03/2017

VICTORIA
State
Government
Environment,
Land, Water
and Planning

Prepared by Roger Fraser, Manager Geodetic Survey

Geodetic Survey
Office of Surveyor-General Victoria
Land Victoria
Department of Environment, Land, Water and Planning
570 Bourke Street
Melbourne VIC 3000
Phone: (03) 8636 2525
Fax: (03) 8636 2776
Email: smes.support@delwp.vic.gov.au
Web: www.delwp.vic.gov.au>Property and land titles>Geodesy

Accessibility

If you would like to receive this publication in an alternative format, please telephone the DELWP Customer Service Centre on 136186, email customer.service@delwp.vic.gov.au or via the National Relay Service on 133 677 www.relayservice.com.au.

Disclaimer

This publication may be of assistance to you but the State of Victoria and its employees do not guarantee that the publication is without flaw of any kind or is wholly appropriate for your particular purposes and therefore disclaims all liability for any error, loss or other consequence which may arise from you relying on any information in this publication.

Contents

1. Introduction

Survey Marks Enquiry Service (SMES) Web services are available for obtaining Permanent Mark (PM) information and/or to submit updates to existing Mark. The API permits searches based upon a location and radius or specific PMs, and can return PM information as basic text, or in the form of PDF, CSV, KML[1] and GeodesyML[2] documents. It also permits authorised user to submit requests to update existing Survey Mark details.

The SMES Web services available in version 2.09.02 include:

- `getMarkInformation` – retrieves PM information as raw data elements. Supports queries using either a list of one or more specific PMs (via nine-figure numbers), or a circular region defined by a centre point and radius.
- `getSurveyMarkSketches` – retrieves PM Sketch Plans as a PDF for one or more specific PMs.
- `getSurveyMarkReports` – retrieves PM Full Reports as a PDF for one or more specific PMs.
- `downloadDocument` – retrieves PM information in the form of CSV, KML and GeodesyML. Supports queries using either a list of one or more specific PMs (via nine-figure numbers), or a circular region defined by a centre point and radius.
- `smesUserLogin` - authenticate user's credentials and supply a sessionKey. Allows authorised user with a valid sessionKey to make Survey Mark update submission(s) with the 'updateMarkDetailsSubmission' method.
- `updateMarkDetailsSubmission` - allows authenticated user to specify a nine figure number and include details to create a submission to update a Survey Mark's details.
- `smesUserLogout` - allows authenticated user with 'smesUserLogin' to logout and invalidate the sessionKey assigned.

These services are designed for ad-hoc requests from desktop and mobile applications.

The Application Programming Interface (API) for the SMES Web services supports two main protocols:

- REST (REpresentational State Transfer). This interface depends upon the standard HTTP GET method and returns data in JavaScript Object Notation (JSON).
- SOAP (Simple Object Access Protocol). This interface sends and receives XML documents to process requests and to return the requested information.

This document describes the SMES Web services APIs using REST and SOAP protocols.

2. Audience

This document is intended for desktop, website and mobile developers who would like to retrieve PM information via custom-developed applications.

3. Contact

For more information, please contact Geodetic Survey via smes.support@delwp.vic.gov.au.

[1] Keyhole Markup Language (KML) is a file format used to display geographic data in an Earth browser such as Google Earth, Google Maps, and Google Maps for mobile.

[2] GeodesyML is a GML application schema developed by the Intergovernmental Committee on Surveying and Mapping (ICSM). It is designed to support the exchange of geodetic information, such as marks, measurements, adjustments and GNSS CORS site log metadata.

4. Web service request / response

The SMES Web services support REST and SOAP access protocols.

4.1. REST Web service request

SMES Web service requests via REST take the following form:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/webService?parameters
```

where ***`webService`*** is the name of SMES Web service to be invoked, and ***`parameters`*** is a collection of optional parameters and values to configure the behaviour of the SMES Web service.

4.2. SOAP Web service request

Calling SMES Web services via SOAP requires the use of XML to specify the Web service to invoke, the parameters and their values. SOAP requests take the following form:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:webService>
         <parameter1></parameter1>
         <parameter2></parameter2>
         <parameter3></parameter3>
         ... etc ...
      </lvis:webService>
   </soapenv:Body>
</soapenv:Envelope>
```

where ***`webService`*** is the name of SMES Web service to be invoked, and ***`parameter1`***, ***`parameter2`***, ***`parameter3`***, ***`... etc ...`*** are the parameter tags holding the values to configure the behaviour of the SMES Web service. The Web Service Description Language (WSDL) which describes the SOAP based Web services is available from:

http://maps.land.vic.gov.au/lvis/services/SmesDataDelivery?wsdl

4.3. Web service response

The response from a SMES Web service request using REST will contain data in JSON. The response from a SMES Web service using SOAP will contain data in XML. On success, the response from both REST and SOAP Web service requests will contain a **`data`** element. On error, the response will contain a **`messages`** element. These elements contain various sub-elements corresponding to each response:

`data` a container for mark information or document. This container holds various sub-elements for each mark or document that is returned. If the response contains a **`data`** element, **`messages`** will not be present. For **`getMarkInformation`**, **`data`** will contain raw data elements relating to each mark. For **`downloadDocument`**, **`getSurveyMarkSketches`** and **`getSurveyMarkReports`**, **`data`** will contain a base 64 binary string representing a single or multi-page document in the required file type.

`messages` a container for any error or exception messages. This container holds one or more message elements that signify the error relating to the request parameters, or if a specified PM nine-figure number does not exist. If the response contains a **`messages`** element, **`data`** will not be present.

5. GetMarkInformation

The **`getMarkInformation`** Web service permits the retrieval of PM information via two techniques:

- Search for PM information using a circular region defined by location and radius
- Request for information for specific PMs using PM nine-figure numbers

The parameters supported by **`getMarkInformation`** are:

`searchType` Mandatory (Text). Signifies the type of search to perform. The available search types are **`Location`** and **`NineFigureNumber`** (provide one only, not both).

`latitude` Optional (Text). The latitude of the centrepoint of the circular region to search for marks. Mandatory if information is requested using `searchType=Location`. Not required for requests using `searchType=nineFigureNumber`. Can be supplied as degrees, minutes and seconds delimited by spaces (e.g. `latitude=37 09 39.23`), or decimal degrees (e.g. `latitude=37.1608972222`). The negative symbol to denote south latitudes is not required.

`longitude` Optional (Text). The longitude of the centrepoint of the circular region to search for marks. Mandatory if information is requested using `searchType=Location`. Not required for requests using `searchType=nineFigureNumber`. Can be supplied as degrees, minutes and seconds delimited by spaces (e.g. `longitude=145 09 39.23`), or decimal degrees (e.g. `longitude=145.1608972222`).

`radius` Optional (Decimal). The radius of the circular region to search for marks. Radius must be >0 and <= 2. Mandatory if information is requested using `searchType=Location`. Not required for requests using `searchType=nineFigureNumber`. Units are in kilometres (e.g. radius=1.25 means 1.25 kilometre radius).

`nineFigureNumber` Optional (Integer). The nine-figure number of the mark for which information is required. Mandatory if information is requested using `searchType=NineFigureNumber`. Not required for requests using `searchType=Location`.

`format` Optional (Text). Signifies the amount of information to return. The available format types are **`Brief`** and **`Full`** (provide one only, not both). If **`format`** is not provided, the parameter will be set to a **`Brief`** when `searchType=Location` and **`Full`** when `searchType= NineFigureNumber`.

`returnDefective` Optional (Boolean). Indicates whether defective marks are to be included in the response. May be supplied when using `searchType=Location` or `searchType= nineFigureNumber`. If **`returnDefective`** is not provided, the parameter will be set to **`false.`**

When searching by location, the required parameters are **`searchType`**, **`latitude`**, **`longitude`** and **`radius`**. When searching by nine-figure numbers, the required parameters are **`searchType`** and **`nineFigureNumber`**. The **`format`** and **`returnDefective`** parameters are optional. The **`format`** parameter signifies the amount of information to return. The **`returnDefective`** parameter signifies that defective marks are to be included in the response.

The information that may be provided in a response from a **`getMarkInformation`** Web service request is shown in Table 1. Note that each response will only contain one element, either the **`data`** container or **`messages`** container (not both). If information for multiple marks is to be returned, the **`data`** container will contain an array of elements.

Table 1 **`getMarkInformation`** response fields

Element		Min	Max	Data Type	Description
`messages`		0	1	Complex	Container for error messages. If this container is present, the data container will not be returned.
	`message`	1	N	Text	The error message.
`data`		0	250	Complex	Container for one or more mark(s). Multiple marks will be handled via an array. If this container is present, the messages container will not be returned.
	`nineFigureNumber`	1	1	Integer	
	`name`	1	1	Text	
	`status`	1	1	Text	
	`scn`	1	1	Boolean (Yes/No)	
	`easting`	1	1	Text	
	`northing`	1	1	Text	
	`zone`	1	1	Integer	
	`latitude`	1	1	Decimal	Values always returned in decimal degrees (e.g. ddd.dddddd)
	`longitude`	1	1	Decimal	Values always returned in decimal degrees (e.g. ddd.dddddd)
	`ahdHeight`	1	1	Text	
	`ellipsoidHeight`	1	1	Text	
	`hUncertainty`	0	1	Text	Returned when `format=Full`
	`vUncertainty`	0	1	Text	Returned when `format=Full`
	`hOrder`	0	1	Text	Returned when `format=Full`
	`vOrder`	0	1	Text	Returned when `format=Full`
	`gda94PublishedDate`	0	1	Date (YYYY-MM-DD)	Returned when `format=Full`
	`gda94Technique`	1	1	Text	
	`gda94Measurements`	0	1	Text	Returned when `format=Full`
	`gda94Source`	0	1	Text	Returned when `format=Full`
	`ahdLevelSection`	0	1	Text	Returned when `format=Full`
	`ahdPublishedDate`	0	1	Date (YYYY-MM-DD)	Returned when `format=Full`
	`ahdTechnique`	1	1	Text	
	`ahdSource`	0	1	Text	Returned when `format=Full`

Specifications on the REST and SOAP protocols are given below.

5.1. REST request

A `getMarkInformation` Web service request via REST takes the following form:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/getMarkInformation?parameters
```

where `parameters` is a collection of mandatory and optional parameters.

Requests via Location and radius

The following examples demonstrate how to retrieve PM information using location and radius. Requests of this nature must use the `searchType`, `latitude`, `longitude` and `radius` parameters.

A REST request using degrees, minutes and seconds for the latitude and longitude:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/getMarkInformation?searchType=Location&la
titude=37 09 39.23&longitude=145 30 19.84&radius=1.1
```

The REST response:

```
{"data":[
{"nineFigureNumber":380300040,"name":"WHANREGARWEN PM
4","status":"OK","scn":"Yes","easting":368070,"northing":5885570,"zone":55,"latitude":-
37.10011,"longitude":145.30505,"ahdHeight":172.862,"ellipsoidHeight":180.672,"gda94Technique":"TRAN
SFORMED","ahdTechnique":"SPIRIT LEVELLING"},
{"nineFigureNumber":380300140,"name":"WHANREGARWEN PM
14","status":"OK","scn":"Yes","easting":367140,"northing":5885220,"zone":55,"latitude":"-
37.10120","longitude":145.30126,"ahdHeight":171.426,"ellipsoidHeight":179.208,"gda94Technique":"TRA
NSFORMED","ahdTechnique":"SPIRIT LEVELLING"}
]}
```

By default, the response will contain brief information. To retrieve all possible information relating to these marks, add `&format=Full` to the request.

A REST request for full information using decimal degrees for the latitude and longitude:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/getMarkInformation?searchType=Location&l
atitude=37.04225&longitude=145.502453&radius=1.1&format=full
```

The REST response:

```
{"data":[{"nineFigureNumber":254600150,"name":"DROPMORE PM
15","status":"OK","scn":"No","easting":366990,"northing":5900200,"zone":55,"latitude":"-
37.02060","longitude":145.30161,"ahdHeight":"","ellipsoidHeight":"","hUncertainty":"","vUncertainty
":"","hOrder":99,"vOrder":"","gda94PublishedDate":"","gda94Technique":"TRANSFORMED","gda94Measureme
nts":"","gda94Source":"OSG","ahdLevelSection":"","ahdPublishedDate":"","ahdTechnique":"","ahdSource
":""},{"nineFigureNumber":254600080,"name":"DROPMORE PM
8","status":"OK","scn":"No","easting":367190,"northing":5899040,"zone":55,"latitude":-
37.02437,"longitude":145.30234,"ahdHeight":"","ellipsoidHeight":"","hUncertainty":"","vUncertainty"
:"","hOrder":99,"vOrder":"","gda94PublishedDate":"","gda94Technique":"TRANSFORMED","gda94Measuremen
ts":"","gda94Source":"OSG","ahdLevelSection":"","ahdPublishedDate":"","ahdTechnique":"","ahdSource"
:""}]}
```

Requests for specific PMs

The following examples demonstrate how to retrieve information for specific PMs using a list of one or more nine-figure numbers. Requests of this nature must use the **type** and **nineFigureNumber** parameters.

A REST request for full information for one mark:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/getMarkInformation?searchType=NineFigure
Number&nineFigureNumber=204610060
```

The REST response:

```
{"data":{"nineFigureNumber":204610060,"name":"BALLARAT PM
1006","status":"OK","scn":"Yes","easting":752952.173,"northing":5840699.371,"zone":54,"latitude":-
37.32456803,"longitude":143.51475928,"ahdHeight":466.062,"ellipsoidHeight":470.858,"hUncertainty":0
.037,"vUncertainty":0.018,"hOrder":3,"vOrder":3,"gda94PublishedDate":"2014-07-
21","gda94Technique":"ADJUSTMENT","gda94Measurements":"DHML","gda94Source":"OSGV","ahdLevelSection"
:"BAL","ahdPublishedDate":"2014-07-14","ahdTechnique":"SPIRIT LEVELLING","ahdSource":"OSGV"}}
```

As shown in the examples above, the `data` container in the REST response from `getMarkInformation` may contain elements for one mark, or an array of elements for multiple marks. Returned values of `""` indicate that there is no data for that field in the database. As the `message` container is only provided on error, in which case a `data` container will not be provided, consuming applications may choose to handle Web service exceptions on the existence of the `message` container.

Errors and exceptions

If a `getMarkInformation` Web service request relates to a mark which is defective and the `returnDefective` parameter has not been provided, or if mark information cannot be retrieved, the response will be as follows:

```
{"messages":{"message":"No survey marks matched the criteria provided."}}
```

If a `getMarkInformation` Web service request doesn't contain a sufficient number of parameters and values (e.g. no `searchType`, `nineFigureNumber`, `latitude`, `longitude` or `radius` parameters), the response will be:

```
{"messages":{"message":"NineFigureNumber search is missing nineFigureNumber."}}
```

or

```
{"messages":{"message":"Location search is missing Latitude, Longitude or Radius."}}
```

If a `getMarkInformation` Web service request contains `nineFigureNumber` or `latitude`, `longitude` and `radius` parameters, but not the mandatory `searchType` parameter, the following response will be returned:

```
{"messages":{"message":"Please provide search type."}}
```

If a `getMarkInformation` Web service request contains `latitude`, `longitude` and `radius` parameters, and the value supplied to `radius` exceeds 2.0 kilometres (e.g. 10 kilometres), the following response will be returned:

```
{"messages":{"message":"The radius supplied '10' is invalid. Radius must be a decimal number
greater than 0 and less than or equal to 2."}}
```

If a **getMarkInformation** Web service request yields more than 250 marks, the following response will be returned:

```
{"messages":{"message":"More than 250 marks were found for this search. Please refine your search
criteria."}}
```

Recommended use for mobile applications

The **getMarkInformation** Web service has been designed to support location based querying using a minimum payload. To maximise the efficiency of this service in a mobile environment, the following request procedure is recommended:

1. Call the **getMarkInformation** Web service with `format=brief`, `searchType=Location` and the **latitude**, **longitude** and **radius** parameters set to the device's location and the desired search radius.

2. From the list of marks returned in the **data** container, form a list of one or more nine-figure numbers (as a comma-delimited string) and then:
    a. Call the **getMarkInformation** Web service with `format=full`, `searchType=NineFigureNumber` and the **nineFigureNumber** parameter set to the list of nine-figure numbers, or;
    b. Call **getSurveyMarkSketches** (see Section 0) or **getSurveyMarkReports** (see Section 0) to obtain PDF sketch plans or full reports respectively.

Depending on user need, Step 2 may or may not be required. For example, the **data** container obtained in Step 1 provides a sufficient amount of information to render marks on a map with symbology (c.f. Section 0), which will readily allow the user to identify Survey Control Network (SCN) marks and non-SCN marks, and if a mark has an AHD height. Step 2 will only be required should the user wish to view detailed information or obtain a PDF Sketch Plan or Report.

5.2. SOAP request

A **getMarkInformation** Web service request via SOAP takes the following form:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:getMarkInformation>
         <searchType></searchType>
         <nineFigureNumber></nineFigureNumber>
         <format></format>
         <returnDefective></returnDefective>
         <latitude></latitude>
         <longitude></longitude>
         <radius></radius>
      </lvis:getMarkInformation>
   </soapenv:Body>
</soapenv:Envelope>
```

where the values for the elements **searchType**, **nineFigureNumber**, **latitude**, **longitude**, **returnDefective** and **radius** are supplied within the respective element tags. The meaning and behaviour of these elements are identical to those parameters required for the RESTful **getMarkInformation** Web service.

Request via Location and radius

The following example demonstrates how to retrieve PM information using location and radius. As with the RESTful approach, requests of this nature must use the **searchType**, **latitude**, **longitude** and **radius** parameters. In this example, decimal degrees is provided for the latitude and longitude and 300 metres for the search radius:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:getMarkInformation>
         <searchType>Location</searchType>
         <latitude>-36.580776</latitude>
         <longitude>146.009925</longitude>
         <radius>0.3</radius>
         <format>Brief</format>
         <returnDefective>false</returnDefective>
      </lvis:getMarkInformation>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getMarkInformationResponse xmlns:ns2="lvis.webservices.smesDataDelivery">
         <return>
            <data>
               <nineFigureNumber>285800110</nineFigureNumber>
               <name>KELFEERA PM   11</name>
               <status>OK</status>
               <scn>Yes</scn>
               <easting>411227.900</easting>
               <northing>5951035.735</northing>
               <zone>55</zone>
               <latitude>-36.34552715</latitude>
               <longitude>146.00277631</longitude>
               <ahdHeight>174.7</ahdHeight>
               <ellipsoidHeight>183.937</ellipsoidHeight>
               <gda94Technique>ADJUSTMENT</gda94Technique>
               <ahdTechnique>UNKNOWN</ahdTechnique>
            </data>
            <data>
               <nineFigureNumber>877160740</nineFigureNumber>
               <name>SR77P74</name>
               <status>OK</status>
               <scn>Yes</scn>
               <easting>411475</easting>
               <northing>5951440</northing>
               <zone>55</zone>
               <latitude>-36.34422</latitude>
               <longitude>146.00379</longitude>
               <ahdHeight>173.180</ahdHeight>
               <ellipsoidHeight>182.419</ellipsoidHeight>
               <gda94Technique>PLOT</gda94Technique>
               <ahdTechnique>SPIRIT LEVELLING</ahdTechnique>
            </data>
         </return>
      </ns2:getMarkInformationResponse>
   </soap:Body>
</soap:Envelope>
```

Note thatas shown above, on success, each response from a SOAP request will contain one or more **`data`** elements.

Requests for specific PMs

The following example demonstrates how to retrieve information for a specific PM using a list of one or more nine-figure numbers. Requests of this nature must use the **type** and **nineFigureNumber** parameters:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:getMarkInformation>
         <searchType>NineFigureNumber</searchType>
         <nineFigureNumber>220901790</nineFigureNumber>
         <format>Full</format>
         <returnDefective>false</returnDefective>
      </lvis:getMarkInformation>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getMarkInformationResponse xmlns:ns2="lvis.webservices.smesDataDelivery">
         <return>
            <data>
               <nineFigureNumber>220901790</nineFigureNumber>
               <name>BOROONDARA PM  179</name>
               <status>OK</status>
               <scn>Yes</scn>
               <easting>330750.127</easting>
               <northing>5808619.069</northing>
               <zone>55</zone>
               <latitude>-37.51150702</latitude>
               <longitude>145.04341745</longitude>
               <ahdHeight>45.315</ahdHeight>
               <ellipsoidHeight>50.21</ellipsoidHeight>
               <hUncertainty>0.035</hUncertainty>
               <vUncertainty>0.100</vUncertainty>
               <hOrder>3</hOrder>
               <vOrder>3</vOrder>
               <gda94PublishedDate>2014-07-21</gda94PublishedDate>
               <gda94Technique>ADJUSTMENT</gda94Technique>
               <gda94Measurements>DHM</gda94Measurements>
               <gda94Source>OSGV</gda94Source>
               <ahdLevelSection/>
               <ahdPublishedDate/>
               <ahdTechnique>SPIRIT LEVELLING</ahdTechnique>
               <ahdSource>MWC</ahdSource>
            </data>
         </return>
      </ns2:getMarkInformationResponse>
   </soap:Body>
</soap:Envelope>
```

Errors and exceptions

On error, a single `messages` container element will be returned, with one or more `message` elements describing the problem. The same messages will be produced as per the RESTful `getMarkInformation` Web service.

6. downloadDocument

The `downloadDocument` Web service permits the retrieval of PM information via two techniques:

- Search for PM information using a circular region defined by location and radius
- Request for information for specific PMs using PM nine-figure numbers

The parameters supported by `downloadDocument` are:

`searchType` Mandatory (Text). Signifies the type of search to perform. The available search types are `Location` and `NineFigureNumber` (provide one only, not both).

`docType` Mandatory (Text). Indicates the type of document to return. The available documents include `CSV`, `KMZ`, and `GML`.

`latitude` Optional (Text). The latitude of the centrepoint of the circular region to search for marks. Mandatory if information is requested using `searchType=Location`. Not required for requests using `searchType=nineFigureNumber`. Can be supplied as degrees, minutes and seconds delimited by spaces (e.g. `latitude=37 09 39.23`), or decimal degrees (e.g. `latitude=37.1608972222`). The negative symbol to denote south latitudes is optional.

`longitude` Optional (Text). The longitude of the centrepoint of the circular region to search for marks. Mandatory if information is requested using `searchType=Location`. Not required for requests using `searchType=nineFigureNumber`. Can be supplied as degrees, minutes and seconds delimited by spaces (e.g. `longitude=145 09 39.23`), or decimal degrees (e.g. `longitude=145.1608972222`).

`radius` Optional (Decimal). The radius of the circular region to search for marks. Radius must be >0 and <= 2. Mandatory if information is requested using `searchType=Location`. Not required for requests using `searchType=nineFigureNumber`. Units are in kilometres (e.g. radius=1.25 means 1.25 kilometre radius).

`marklist` Optional (Text). A comma separated list (without spaces) of one or more nine-figure numbers for which information is to be retrieved. Mandatory if information is requested using `searchType=NineFigureNumber`

`returnDefective` Optional (Boolean). Indicates whether defective marks are to be included in the response. May be supplied when using `searchType=Location` or `searchType= nineFigureNumber`. If `returnDefective` is not provided, the parameter will be set to `false.`

When searching by location, the required parameters are `searchType`, `docType`, `latitude`, `longitude` and `radius`. When searching by nine-figure numbers, the required parameters are `searchType`, `docType` and `markList`. The `format` and `returnDefective` parameters are optional. The `returnDefective` parameter signifies that defective marks are to be included in the response.

The information that may be provided in a response from a `downloadDocument` Web service request is shown in Table 2. If information for multiple marks is to be returned, the `document` container will contain multiple marks within the base64 string. Note – either the `data` container or `messages` container (not both) will be returned for each response.

Table 2 **downloadDocument** response fields

Element	Min	Max	Data Type	Description
messages	0	1	Complex	Container for error messages. If this container is present, the data container will not be returned.
message	1	N	Text	The error message.
data	0	1	Complex	Container for one or more mark document(s). Multiple marks will be handled via an array. If this container is present, the messages container will not be returned.
document	1	1	Text	A base64 binary string representing the document in the requested file type
documentType	1	1	Text	One of CSV, KMZ or GML (being GeodesyML)

Specifications on the REST and SOAP protocols are given below.

6.1. REST request

A **downloadDocument** Web service request via REST takes the following form:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/downloadDocument?parameters
```

where **parameters** is a collection of mandatory and optional parameters. On success, the response from a **downloadDocument** Web service request will be represented in the following form:

```
{"data":{"document":"... base64 binary string ...", "documentType":"XXX"}}
```

The value in the **document** element will be a base64 binary string representing the document in the requested file type. The value in the **documentType** element (shown above as “XXX”) will be a string corresponding to the requested document type, which may be one of "CSV", "GML" or “KMZ”. If a KML document is required, the request must be KMZ, and the response will always be a KMZ containing the KML and png image(s) corresponding to the mark types (c.f. Section 0).

Requests via Location and radius

The following examples demonstrate the REST request and response for retrieving PM information as CSV, KMZ and GeodesyML documents using location and radius. Requests of this nature must use the **searchType**, **docType**, **latitude**, **longitude** and **radius** parameters. The **returnDefective** parameter is only required if defective marks are to be included in the response.

A REST request for a CSV document using degrees, minutes and seconds for the latitude and longitude and a 1.5 kilometre search radius:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/downloadDocument?searchType=Location&docType=csv&latitude=37%2009%2039.23&longitude=145%2030%2019.84&radius=1.5
```

The REST response:

```
{"data":{"document":"U01FUyBTZWFyY2ggcmVzdWx0cy4gR2VuZXJhdGVkIGRhdGU6IDE2LzA5LzIwMTUKTWFyayBJRCw5IE
ZpZ3VyZSBObyxOYW1lLFNDTixFYXN0aW5nLE5vcnRoaW5nLFpvbmUsTGF0aXR1ZGUgKGRkLm1tc3NzcyksTG9uZ2l0dWRlIChkZ
...
JNRUQsTCxPU0csMTc1LjEyMywsLFNQSVJJVCBMRVZFTExJTkcsUldDLAo=","documentType":"CSV"}}
```

A REST request for a KML document using decimal degrees for the latitude and longitude and a 1.25 kilometre search radius:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/downloadDocument?searchType=Location&doc
Type=kmz&latitude=37.1608972222&longitude=145.5055111111&radius=1.25
```

The REST response:

```
{data":{"document":"UEsDBBQACAgIAAxQMEcAAAAAAAAAAAAAAAAAfAAAAMjAxNS0wOS0xNi0xMC0wMC1zbWVzLW1hcmtzLmt
tbO2bW3OiSBTH3\/Mpupyq8WWF5g5GnXKjSawxmpJcavcNoVepQXDoNppvvw3eEdPIbHbiDF15gNPnwJ\/T0P44TWpfFhMPvKAQ
...
FBgAAAAADAAMA1wAAAFUJAAAAAA==","documentType":"KMZ"}}
```

A REST request for a GeodesyML document (using decimal degrees for the latitude and longitude):

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/downloadDocument?searchType=Location&docT
ype=gml&latitude=37.1608972222&longitude=145.5055111111&radius=1.25
```

The REST response:

```
{"data":{"document":"PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0ndXRmLTgnPz4KPGdlbzpHZW9kZXN5TUwgeG1sbn
M6Z2VvPSJ1cm46eG1sLWdvdi1hdTppY3NtOmVnZW9kZXN5OjAuMSIgeG1sbnM6Z21sPSJodHRwOi8vd3d3Lm9wZW5naXMubmV0L
...
RtZW50Pgo8L2dlbzpHZW9kZXN5TUw+Cg==","documentType":"GML"}}
```

Requests for specific PMs

The following examples demonstrate the REST request and response for retrieving information for specific PMs using a list of one or more nine-figure numbers. Requests of this nature must use the **searchType**, **docType** and **marklist** parameters. The **returnDefective** parameter is only required if defective marks are to be included in the response.

A REST request for three marks in a CSV document:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/downloadDocument?searchType=nineFigureNu
mber&docType=csv&markList=260500230,260500200,260500220
```

The REST response:

```
{"data":{"document":"U01FUyBTZWFyY2ggcmVzdWx0cy4gR2VuZXJhdGVkIGRhdGU6IDE2LzA5LzIwMTUKTWFyayBJRCw5IE
ZpZ3VyZSBObyxOYW1lLFNDTixFYXN0aW5nLE5vcnRoaW5nLFpvbmUsTGF0aXR1ZGUgKGRkLm1tc3NzcyksTG9uZ2l0dWRlIChkZ
...
0ssLFRSQVZFUlNFLCwsLCwsLCwK","documentType":"CSV"}}
```

A REST request for one mark in a KML document:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/downloadDocument?searchType=nineFigureNu
mber&docType=kmz&markList=260500230
```

The REST response:

```
{"data":{"document":"UEsDBBQACAgIAIFRMEcAAAAAAAAAAAAAAAAfAAAAMjAxNS0wOS0xNi0xMC0xMi1zbWVzLW1hcmtzLm
ttbLVX23KbOhR9z1do6Ez9cszVdgwBOm7tNplTO5mQ9KFvMuiApiBRSY6dvz9CtoONaXEv0ZNYey1psXVh47\/bFDl4QoxjSgLN
...
ACAJUAAAAGBQAAAAA=","documentType":"KMZ"}}
```

A REST request for three marks in a GeodesyML document:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/downloadDocument?searchType=nineFigureNu
mber&docType=gml&markList=865200800,360500080,201500410
```

The REST response:

```
{{"data":{"document":"PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0ndXRmLTgnPz4KPGdlbzpHZW9kZXN5TUwgeG1sb
nM6Z2VvPSJ1cm46eG1sLWdvdi1hdTppY3NtOmVnZW9kZXN5OjAuMSIgeG1sbnM6Z21sPSJodHRwOi8vd3d3Lm9wZW5naXMubmV0
...
ZW86R2VvZGVzeU1MPgo=","documentType":"GML"}}}
```

Errors and exceptions

If a **downloadDocument** Web service request relates to a mark which is defective and the **returnDefective** parameter has not been provided, or if mark information cannot be retrieved, the response will be as follows:

```
{"messages":{"message":"No survey mark information found for given input"}}
```

If a **downloadDocument** Web service request doesn't contain a sufficient number of parameters and values (e.g. no **searchType**, **docType**, **marklist**, **latitude**, **longitude** or **radius** parameters), or yields no results (e.g. a specified mark does not exist, or no marks exist within the circular region), the response will be:

```
{"messages":{"message":"NineFigureNumber search is missing nineFigureNumber."}}
```

or

```
{"messages":{"message":"Location search is missing Latitude, Longitude or Radius."}}
```

If a **downloadDocument** Web service request does not contain the mandatory **docType** parameter, the following response will be returned:

```
{"messages":{"message":"Please provide document type"}}
```

If a **downloadDocument** Web service request contains **latitude**, **longitude** and **radius** parameters, and the value supplied to **radius** exceeds 2.0 kilometres (e.g. 20 kilometres), the following response will be returned:

```
{"messages":{"message":"The radius supplied '20' is invalid. Radius must be a decimal number
greater than 0 and less than or equal to 2."}}
```

If a `downloadDocument` Web service request yields more than 250 PMs, the following response will be returned:

```
{"messages":{"message":"More than 250 marks were found for this search. Please refine your search
criteria."}}
```

6.2. SOAP request

A `downloadDocument` Web service request via SOAP takes the following form:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesDataDelivery">
  <soapenv:Header/>
  <soapenv:Body>
    <lvis:downloadDocument>
      <searchType></searchType>
      <docType></docType>
      <marklist></marklist>
      <returnDefective></returnDefective>
      <latitude></latitude>
      <longitude></longitude>
      <radius></radius>
    </lvis:downloadDocument>
  </soapenv:Body>
</soapenv:Envelope>
```

where the values for the named elements `searchType`, `docType`, `marklist`, `returnDefective`, `latitude`, `longitude` and `radius` are supplied within the element tags as required. The meaning and behaviour of these elements are identical to those parameters required for the RESTful `downloadDocument` Web service.

The namespace for the SOAP reponse from a `downloadDocument` Web service request will be named `downloadDocumentResponse`.

Request via Location and radius

The following example demonstrates a SOAP request and response for retrieving PM information in a GeodesyML document using location and radius. As with the RESTful approach, requests of this nature must use the `searchType`, `docType`, `latitude`, `longitude` and `radius` XML element tags. In this example, decimal degrees is provided for the latitude and longitude and 1.5 kilometres for the search radius:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:downloadDocument>
         <searchType>Location</searchType>
         <docType>GML</docType>
         <latitude>37.1608972222</latitude>
         <longitude>146.009925</longitude>
         <radius>1.5</radius>
      </lvis:downloadDocument>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:downloadDocumentResponse xmlns:ns2="lvis.webservices.smesDataDelivery">
         <return>
            <data>
<document>PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0ndXRmLTgnPz4KPGdlbzpHZW9kZXN5TUwgeG1sbnM6Z2VvPSJ1c
m46eG1sLWdvdi1hdTppY3NtOmVnZW9kZXN5OjAuMSIgeG1sbnM6Z21sPSJodHRwOi8vd3d3Lm9wZW5naXMubmV0L2dtbC8zLjIi
...
poYXNNb251bWVudCB4bGluazpocmVmPSIjaWQzIi8+CjwvZ2VvOk5vZGU+CjwvZ2VvOkdlb2Rlc3lNTD4K</document>
               <documentType>GML</documentType>
            </data>
         </return>
      </ns2:downloadDocumentResponse>
   </soap:Body>
</soap:Envelope>
```

Requests for specific PMs

The following example demonstrates how to retrieve information for 3 PMs in CSV format. Requests of this nature must use the **searchType**, **docType** and **marklist** XML elements.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:downloadDocument>
         <searchType>NineFigureNumber</searchType>
         <docType>CSV</docType>
         <markList>865200800,360500080,201500410</markList>
      </lvis:downloadDocument>
   </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:downloadDocumentResponse xmlns:ns2="lvis.webservices.smesDataDelivery">
         <return>
            <data>
<document>U01FUyBTZWFyY2ggcmVzdWx0cy4gR2VuZXJhdGVkIGRhdGU6IDE4LzA5LzIwMTUKTWFyayBJRCw5IEZpZ3VyZSBOb
yxOYW1lLFNDTixFYXN0aW5nLE5vcnRoaW5nLFpvbmUsTGF0aXR1ZGUgKGRkLm1tc3NzcyksTG9uZ2l0dWRlIChkZC5tbXNzc3Mp
...
wzLE9LLCxQTE9ULEwsR0VPRCw4NC4xODcsLCxTUElSSVQgTEVWRUxMSU5HLFJXQywK</document>
               <documentType>CSV</documentType>
            </data>
         </return>
      </ns2:downloadDocumentResponse>
   </soap:Body>
</soap:Envelope>
```

Errors and exceptions

On error, a single **messages** container element will be returned, with one or more **message** elements describing the problem. The same messages will be produced as per the RESTful **downloadDocument** Web service.

7. getSurveyMarkSketches

The **getSurveyMarkSketches** Web service permits the retrieval of specific PM Sketch Plans. The parameters supported by **getSurveyMarkSketches** are:

marklist Mandatory (Text). A comma separated list (without spaces) of one or more nine-figure numbers for which PM Sketch Plans are required.

returnDefective Optional (Boolean). Indicates whether defective marks are to be included in the response.

The information that may be provided in a response from a **getSurveyMarkSketches** Web service request is shown in Table 3. If information for multiple marks is to be returned, the **document** container will contain multiple marks within the base64 string. Note – either the **data** container or **messages** container (not both) will be returned for each response.

Table 3 **getSurveyMarkSketches** response fields

Element		Min	Max	Data Type	Description
messages		0	1	Complex	Container for error messages. If this container is present, the data container will not be returned.
	message	1	N	Text	The error message.
data		0	1	Complex	Container for one or more mark document(s). Multiple marks will be handled via an array. If this container is present, the messages container will not be returned.
	document	1	1	Text	A base64 binary string representing the PDF document
	nineFigureNumber	1	1	Integer	The nine-figure number for this document.

Specifications on the REST and SOAP protocols are given below.

7.1. REST request

A **getSurveyMarkSketches** Web service request via REST takes the following form:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/getSurveyMarkSketches?marklist=value
```

where the value supplied to **marklist** is a comma separated list (without spaces) of one or more PM nine-figure numbers for which PM Sketch Plans are required. On success, the response from a **getSurveyMarkSketches** Web service request for a single mark will be represented in the following form:

```
{"data":{"document":"... base64 binary stream ... ","nineFigureNumber":NNNNNNNNN}}
```

The **document** element in the REST response from **getSurveyMarkSketches** will be a binary string representing a single or multipage PDF document. If a PM Sketch Plan is requested for a single mark and that mark has multiple sketches, the returned PDF document will be a multipage PDF document containing all Sketch Plan images. The **nineFigureNumber** element will contain an integer representing the nine-figure number for the mark.

If PM Sketch Plans for multiple marks are requested (via a comma delimited string in **marklist**), the **data** element will contain an array of **document** elements. Arrays are represented in JSON by a comma-separated list encapsulated by square brackets. As an example, information for two marks will be returned as follows:

```
{"data":[
  {"document":"JVBERi0xLjQKJeLjz9MKMiA ... 0NgolJUVPRgo=","nineFigureNumber":209500280},
  {"document":"JVBERi0xLjQKJeLjz9MKMiA ... xMzEKJSVFT0YK","nineFigureNumber":209500290}
]}
```

Example

The following example demonstrates a REST request and response for PM Sketch Plan(s) for a single mark:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/getSurveyMarkSketches?markList=204611920
```

The REST response:

```
{"data":{"document":"JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9MZW5ndGggNjYvRmlsdGVyL0ZsYXRlRGVjb2RlPj5zdH
JlYW0KeJwr5CrkcgrhMlQwAEJDBVNDSz0zIwUjA4WQXC79iEwjcwULhZA0Lo2AxPRUoHx+moKhZkgWl2sIVyAQAgCflw2WCmVuZ
...
SRlDehIsygxJGQsyCcpClRU9GCg==","nineFigureNumber":204611920}}
```

Errors and exceptions

If a **getSurveyMarkSketches** Web service request relates to a mark which is non-existent (e.g. 123456789), the response will be as follows:

```
{"messages":{"message":"Nine figure number '123456789' does not exist."}}
```

7.2. SOAP request

A **getSurveyMarkSketches** Web service request via SOAP takes the following form (e.g. using nine-figure numbers 220901790, 106950099 and 220902150):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:getSurveyMarkSketches>
         <markList>220901790,106950099,220902150</markList>
      </lvis:getSurveyMarkSketches>
   </soapenv:Body>
</soapenv:Envelope>
```

As shown above, the value supplied within the **marklist** element tags is a comma separated list (without spaces) of one or more PM nine-figure numbers for which PM Sketch Plans are required.

The response for this request is:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getSurveyMarkSketchesResponse xmlns:ns2="lvis.webservices.smesDataDelivery">
         <return>
            <data>
               <document>JVBERi0xLjQKJeLjz9MK . . . 0eHJlZgo1MjY1NwolJUVPRgo=</document>
               <nineFigureNumber>106950099</nineFigureNumber>
            </data>
            <data>
               <document>JVBERi0xLjQKJeLjz9MK . . . 4cmVmCjE1OTI1NgolJUVPRgo=</document>
               <nineFigureNumber>220901790</nineFigureNumber>
            </data>
            <data>
               <document>JVBERi0xLjQKJeLjz9MK . . . 4cmVmCjkxMTgzCiUlRU9GCg==</document>
               <nineFigureNumber>220902150</nineFigureNumber>
            </data>
         </return>
      </ns2:getSurveyMarkSketchesResponse>
   </soap:Body>
</soap:Envelope>
```

The namespace for the SOAP reponse from a **getSurveyMarkSketches** Web service request will be named **getSurveyMarkSketchesResponse**. The meaning and behaviour of these elements are identical to those parameters required for the RESTful **getSurveyMarkSketches** Web service.

Errors and exceptions

On error, a single **messages** container element will be returned, with one or more **message** elements describing the error. The same messages will be produced as per the RESTful **getSurveyMarkSketches** Web service.

8. getSurveyMarkReports

The **getSurveyMarkReports** Web service permits the retrieval of PM Full Reports. The parameters supported by **getSurveyMarkReports** are:

marklist	Mandatory (Text). A comma separated list (without spaces) of one or more nine-figure numbers for which PM Full Reports are required.
returnDefective	Optional (Boolean). Indicates whether defective marks are to be included in the response.

The information that may be provided in a response from a **getSurveyMarkReports** Web service request is shown in Table 4. Note – either the **data** container or **messages** container (not both) will be returned for each response.

Table 4 **getSurveyMarkReports** response fields

Element	Min	Max	Data Type	Description
`messages`	0	1	Complex	Container for error messages. If this container is present, the data container will not be returned.
`message`	1	N	Text	The error message.
`data`	0	1	Complex	Container for one or more mark document(s). Multiple marks will be handled via an array. If this container is present, the messages container will not be returned.
`document`	1	1	Text	A base64 binary string representing the PDF document
`nineFigureNumber`	1	1	Integer	The nine-figure number for this document.

The **document** element in the REST response from **getSurveyMarkReports** will be a binary string representing a single PDF document. The **nineFigureNumber** element will contain an integer representing the nine-figure number for the mark. If a PM Report for multiple marks are requested (via a comma delimited string in **marklist**), the **data** element will contain an array of **document** elements.

Specifications on the REST and SOAP protocols are given below.

8.1. REST request

A **getSurveyMarkReports** Web service request via REST takes the following form:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/getSurveyMarkReports?marklist=value
```

where the value supplied to **marklist** is a comma separated list (without spaces) of one or more PM nine-figure numbers for which PM Full Reports are required. On success, the response from a **getSurveyMarkReports** Web service request for a single mark will be represented in the following form:

```
{"data":{"document":"... base64 binary stream ... ","nineFigureNumber":NNNNNNNNN}}
```

The `document` element in the REST response from `getSurveyMarkReports` will be a binary string representing a single PDF document. The `nineFigureNumber` element will contain an integer representing the nine-figure number for the mark. If PM Full Reports for multiple marks are requested (via a comma delimited string in `marklist`), the `data` element will contain an array of `document` elements.

Example

The following example demonstrates a REST request and response for a PM Full Report for two marks:

```
http://maps.land.vic.gov.au/lvis/services/smesDataDelivery/getSurveyMarkReports?markList=367100170,
315000140
```

The REST response containing two PM Full Reports (represented by a JSON array):

```
{"data":[
  {"document":"JVBERi0xLjQKJeLjz9M . . . MKJSVFT0YK","nineFigureNumber":367100170},
  {"document":"JVBERi0xLjQKJeLjz9M . . . olJUVPRgo=","nineFigureNumber":315000140}
]}
```

Errors and exceptions

If a `getSurveyMarkReports` Web service request relates to a mark which is non-existent (e.g. 123456789), the response will be as follows:

```
{"messages":{"message":"Nine figure number '123456789' does not exist."}}
```

8.2. SOAP request

A `getSurveyMarkReports` Web service request via SOAP takes the following form (e.g. using nine-figure numbers 220901790, 106950099 and 220902150):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:getSurveyMarkReports>
         <markList>383400370,253900300</markList>
      </lvis:getSurveyMarkReports>
   </soapenv:Body>
</soapenv:Envelope>
```

As shown above, the value supplied within the `marklist` element tags is a comma separated list (without spaces) of one or more PM nine-figure numbers for which PM Full Reports are required.

The response for this request is:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getSurveyMarkReportsResponse xmlns:ns2="lvis.webservices.smesDataDelivery">
         <return>
            <data>
               <document>JVBERi0xLjQKJeLjz9M . . . hyZWYKNjI4MzEKJSVFT0YK</document>
               <nineFigureNumber>383400370</nineFigureNumber>
            </data>
            <data>
               <document>JVBERi0xLjQKJeLjz9M . . . JlZgo2MjgwOQolJUVPRgo=</document>
               <nineFigureNumber>253900300</nineFigureNumber>
            </data>
         </return>
      </ns2:getSurveyMarkReportsResponse>
   </soap:Body>
</soap:Envelope>
```

The namespace for the SOAP reponse from a **getSurveyMarkReports** Web service request will be named **getSurveyMarkReportsResponse**.

Errors and exceptions

On error, a single **messages** container element will be returned, with one or more **message** elements describing the error. The same messages will be produced as per the RESTful **getSurveyMarkReports** Web service.

9. smesUserLogin

The **smesUserLogin** Web service authenticate a user's credentials and supply a sessionKey. User with a valid sessionKey is able to lodge Survey Mark update submission(s) with the **updateMarkDetailsSubmission** web service.

The parameters supported by **smesUserLogin** are:

Element	Min	Max	Data Type	Description
userName	1	1	Text	User name of an authorised SMES user.
password	1	1	Text	Password of the authorised SMES user.

The response returned by **smesUserLogin** are:

Element	Min	Max	Data Type	Description
messages	0	1	Complex	Container for any errors. If <messages> is present, <data> will not be present.
Message	1	1	Text	Error message.
data	0	1	Complex	Container for the response. If <data> is present, <messages> will not be present.
sessionKey	1	1	Text	Contains sessionKey to be used for the 'updateMarkDetailsSubmission' method.

N.B.: Min = 0 indicates an optional field. Min = 1 indicates a mandatory field.

9.1. REST request

A **smesUserLogin** Web service request via REST takes the following form:

```
https://maps.land.vic.gov.au/lvis/services/smesSurveyMarkDataDelivery/smesUserLogin?userName=value&password=value
```

Example

```
https://maps.land.vic.gov.au/lvis/services/smesSurveyMarkDataDelivery/smesUserLogin?userName=OSGVuser&password=password
```

The REST response:

```
{"data":{"sessionKey":"-1W6VK8Z1DKB47|24-02-2017"}}
```

Errors and exceptions

If a **`smesUserLogin`** Web service request failed as no parameters were specified, the response will be as follows:

```
{"messages":{"message":"Username and Password are mandatory, please try again."}}
```

If the userName and password does not match or the login was unsuccessful or if the system was unable to conncet to MyProfile to perform user authentication, the response will be as follows:

```
{"messages":{"message":" Authentication failed, please try again."}}
```

If the user with a valid sessionKey calls the **`smesUserLogin`** Web service again, the response will be as follows:

```
{"messages":{"message":" User is already logged in."}}
```

9.2. SOAP request

A **`smesUserLogin`** Web service request via SOAP takes the following form (using username, password):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesSurveyMarkDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:smesUserLogin>
         <userName>osgvuser</userName>
         <!--Optional:-->
         <password>password</password>
      </lvis:smesUserLogin>
   </soapenv:Body>
</soapenv:Envelope>
```

The response for this request is:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:smesUserLoginResponse xmlns:ns2="lvis.webservices.smesSurveyMarkDataDelivery">
         <return>
            <data>
               <sessionKey>-1FI5GCEFJD4IF|23-02-2017</sessionKey>
            </data>
         </return>
      </ns2:smesUserLoginResponse>
   </soap:Body>
</soap:Envelope>
```

If the Login was successfully, a **sessionKey** will be allocated and remains active for 2 minutes. The sessionKey expires after 2 minutes of inactivity. It remains active if used continuously before expiry.

If a request completes within 2 minutes, then the user is able to submit a new request with the same sessionKey.

Errors and exceptions

On error, a single **`messages`** container element will be returned. The same messages will be produced as per the RESTful **`smesUserLogin`** Web service.

10. smesUserLogout

The **smesUserLogout** Web service logout an authenticated user and invalidates their sessionKey.

The parameters supported by **smesUserLogout** are:

Element	Min	Max	Data Type	Description
sessionKey	1	1	Text	The sessionKey of current session.

The response returned by **smesUserLogout** are:

Element	Min	Max	Data Type	Description
messages	0	1	Complex	Container for any errors. If <messages> is present, <data> will not be present.
Message	1	1	Text	Error message.
data	0	1	Complex	Container for the response. If <data> is present, <message> will not be present.
sessionKey	1	1	Text	Session cessation message.

N.B.: Min = 0 indicates an optional field. Min = 1 indicates a mandatory field.

10.1. REST request

A **smesUserLogout** Web service request via REST takes the following form:

```
https://maps.land.vic.gov.au/lvis/services/smesSurveyMarkDataDelivery/smesUserLogout?sessionKey=
value
```

Example

```
https://maps.land.vic.gov.au/lvis/services/smesSurveyMarkDataDelivery/smesUserLogout
?sessionKey=-130KZU6ERYPL1|24-02-2017
```

The REST response:

```
{"messages":{"message":"Session terminated successfully."}}
```

Errors and exceptions

If a **smesUserLogout** Web service request failed if invalid sessionKey was specified, the response will be as follows:

```
{"messages":{"message":" Invalid sessionKey."}}
```

If the userName and password does not match or the login was unsuccessful or if the system was unable to conncet to MyProfile to perform user authentication, the response will be as follows:

```
{"messages":{"message":" Authentication failed, please try again."}}
```

If the user with a valid sessionKey calls the **smesUserLogin** Web service again, the response will be as follows:

```
{"messages":{"message":" User is already logged in."}}
```

10.2. SOAP request

A **smesUserLogin** Web service request via SOAP takes the following form (using username, password):

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesSurveyMarkDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:smesUserLogout>
         <sessionKey>-1Q9Z59D4WDTMU|23-02-2017</sessionKey>
      </lvis:smesUserLogout>
   </soapenv:Body>
</soapenv:Envelope>
```

The response for this request is:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:smesUserLogoutResponse xmlns:ns2="lvis.webservices.smesSurveyMarkDataDelivery">
         <return>
            <messages>
               <message>Session terminated successfully.</message>
            </messages>
         </return>
      </ns2:smesUserLogoutResponse>
   </soap:Body>
</soap:Envelope>
```

If the request was successful, the user’s session will be terminated, the sessionKey will be invalidated and a message ‘Session terminated successfully.’ is returned.

Errors and exceptions

On error, a single **messages** container element will be returned. The same messages will be produced as per the RESTful **smesUserLogout** Web service.

11. updateMarkDetailsSubmission

The `updateMarkDetailsSubmission` Web service allows authenticated user to submit entries to update existing Survey Mark details.

The parameters supported by `updateMarkDetailsSubmission` are:

Element	Min	Max	Data Type	Description
sessionKey	1	1	Text	The unique sessionKey that the system has generated for this user in the current session.
nineFigureNumber	1	1	Integer	The nine figure number of an existing Survey Mark to be updated.
markStatus	0	1	Text	Optional fields. If present, must be one of the following values: • Abandoned • Damaged • Destroyed • Disturbed • Nipple Damaged • Not Found • Not Used • OK • Plaque Missing • Removed • Replaced • Suspect • Unknown • Unstable
planNumber	0	1	Text	Optional field.
groundToMarkOffset	0	1	Decimal	Optional field. If present, must be a value between -99.999 and 99.999.
coverExists	0	1	Text	Optional fields. If present, must be one of the following values: • Yes • No
markerPostExists	0	1	Text	Optional fields. If present, must be one of the following values: • Yes • No
gnssSuitability	0	1	Text	Optional fields. If present, must be one of the following values: • Good • Moderate • Poor
markType	0	1	Text	Optional fields. If present, must be one of the following values: • Plaque • Cross Head Nail • Deep Driven Rod • Pipe • Rivet • Star Picket • Other
comments	1	1	Text	Comments to be saved against the updated survey mark. Max 1000 characters supported. Any excess will be truncated.

Element	Min	Max	Data Type	Description
supportingFiles	0	1	Complex	Container for optional file attachment(s).
files	1	10	Text	This section repeats if > 1 files were attached. Max 10 files. Max 25MB total file size.
file	1	1	Text	Attached file (must be in a base-64 encoded byte stream). Max 5 MB per file.
fileName	1	1	Text	File name of the attached file
fileExtension	1	1	Text	File extension of the attached file. (see below)
fileCategory	1	1	Text	Optional fields. If present, must be of the following values: • Sketch Plan • Mark Photograph • Raw Survey Data • Other

N.B.: Min = 0 indicates an optional field. Min = 1 indicates a mandatory field. Each time a validated user calls this method, the sessionKey timeout is reset (2 minutes).

No.	fileCategory	fileExtension
1	Sketch Plan	pdf
2	Mark Photograph	jpg or jpeg
3	Raw Survey Data	t01, lb2, zip, yyo or dat
4	Other	doc, docx, xls, xlsx, csv, png, bmp and txt

N.B.: Please note that for the file extension **yyo** for **RINEX** files where the 'yy' denotes **numeric** numbers. E.g. if the RINEX file is from 2016, then the uploaded file extension should be '.16o'. However SMES will only validate yy as a numeric number but will not validate it against the actual file content vs its extension.

The response returned by `updateMarkDetailsSubmission` are:

Element	Min	Max	Data Type	Description
messages	0	1	Complex	Container for any errors. If <messages> is present, <data> will not be present.
Message	1	1	Text	Error message.
data	0	1	Complex	Container for the response. If <data> is present, <message> will not be present.
sessionKey	1	1	Text	Contains confirmation when mark details have been successfully processed and/or any exception message(s).

11.1. REST request

A **`updateMarkDetailsSubmission`** Web service request via REST takes the following form:

```
POST
https://maps.land.vic.gov.au/lvis/services/smesSurveyMarkDataDelivery/updateMarkDetailsSubmission

REQUEST BODY
{"sessionKey":<value>,
"nineFigureNumber":<value>,
"markStatus ":<value>,
"planNumber ":<value>,
"groundToMarkOffset":<value>,
"coverExists":<value>,
"markerPostExists":<value>,
"gnssSuitability":<value>,
"markType":<value>,
"comments":<value>,
"supportingFiles":{
      "files":[ {
        "file": <base 64 encoded byte stream>,
        "fileName": <value>,
        "fileExtension": <value>,
        "fileCategory": <value>
      }]
}}
```

Example

Full parameters

```
POST
https://maps.land.vic.gov.au/lvis/services/smesSurveyMarkDataDelivery/updateMarkDetailsSubmis
sion

REQUEST BODY
{"sessionKey":-1LPV08R3K13II|07-03-2017,
"nineFigureNumber":<<ninefigurenumber>>,
"markStatus":"Abandoned",
"planNumber":"AL1112",
"groundToMarkOffset":-23.231,
"coverExists":"No",
"markerPostExists":"No",
"gnssSuitability":"Moderate",
"markType":"Cross Head Nail",
"comments":"Update from REST Services",
"supportingFiles":{
      "files":[ {
        "file": "c3RyaW5n",
        "fileName": "Webservice_test_file",
        "fileExtension": "pdf",
        "fileCategory": "Sketch Plan"
      }],
"files":[{
        "file": "aGkgdGhlcmUgaWFtIHRlc3RpbmcgY3I0NTkyIHdpdGggbXVsdGlwbGUgZmlsZXM=",
        "fileName": "Ala-3-Other",
        "fileExtension": "docx",
        "fileCategory": "Other"
      }],
"files":[{
        "file": "aGkgdGhlcmUgaWFtIHRlc3RpbmcgY3I0NTkyIHdpdGggbXVsdGlwbGUgZmlsZXM=",
        "fileName": "Ala-1-Raw",
        "fileExtension": "zip",
        "fileCategory": "Raw Survey Data"
     }],
"files":[{
        "file": "aGkgdGhlcmUgaWFtIHRlc3RpbmcgY3I0NTkyIHdpdGggbXVsdGlwbGUgZmlsZXM=",
        "fileName": "Ala-2-Mark",
        "fileExtension": "JPG",
        "fileCategory": "Mark Photograph"
      }]
    } }
```

Only mandatory parameters

```
POST
https://maps.land.vic.gov.au/lvis/services/smesSurveyMarkDataDelivery/updateMarkDetailsSubmission

REQUEST BODY
{"sessionKey":-1RNTJBWCETXDL|24-02-2017,
"nineFigureNumber":254105580,
"comments":"Update from REST Services"}
```

The REST response:

```
{"messages": {"message": "Survey Mark successfully submitted for approval."}}
```

Errors and exceptions

A **updateMarkDetailsSubmission** Web service request fails if invalid sessionKey was specified, the response will be as follows:

```
{"messages":{"message":"Invalid sessionKey."}}
```

A **updateMarkDetailsSubmission** Web service request fails if mandatory fields were not specified, the response will be as follows:

```
{"messages":{"message":"Mandatory fields: sessionKey, nineFigurenumber and comments are
required."}}
```

A **updateMarkDetailsSubmission** Web service request fails if the specified 9 figure number of an existing survey mark could not be found, the response will be as follows:

```
{"messages":{"message":"Survey mark not found."}}
```

A **updateMarkDetailsSubmission** Web service request fails due to internal system error(s), the response will be as follows:

```
{"messages":{"message":"Survey mark update submission failed due to internal error(s). Please try
again."}}
```

A **updateMarkDetailsSubmission** Web service request fails when there is/are error(s) in any file attribute(s) (the ‘file’, ‘fileName’, ‘fileExtension’, ‘fileCategory’ parameter), the response will be as follows:

A file was attached but fileName/ fileExtension/ fileCategory was not provided

```
{"messages":{"message":"<file name provided>/A file cannot be attached as fileName or
fileExtension or fileCategory was not supplied."}}
```

A file was attached but fileCategory and corresponding fileExtension does not match

```
{"messages":{"message":"<file name provided>/A file cannot be attached as fileExtension does
not match with the fileCategory."}}
```

A file was attached but fileCategory has invalid value

```
{"messages":{"message":"<file name provided>/A file cannot be attached as the fileCategory is
invalid."}}
```

A file was not accepted because it is larger than 5MB

```
{"messages":{"message":"<file name provided>/A file cannot be attached as the file size is
larger than 5MB."}}
```

A file was not accepted because the total file size exceeds 25MB

```
{"messages":{"message":"<file name provided>/A file cannot be attached as the total file size
is larger than 25MB."}}
```

Miscellaneous errors (e.g. decode of byte stream failed)

```
{"messages":{"message":"<file name provided>/A file cannot be attached due to internal
error(s)."}}
```

If a `updateMarkDetailsSubmission` Web service request was submitted with option field(s) having invalid value, an exception message will be appended and returned:

```
{"messages":{"message":"<field name> has invalid value of <field value>. This value is ignored
for submission."}}
```

If a `updateMarkDetailsSubmission` Web service request was submitted with 1 or multiple **exception message(s),** the response will be as follows:

```
{"messages":{"message":"Survey mark details submitted for approval with the following exceptions.
<exception message(s)>"}}
```

11.2. SOAP request

A **updateMarkDetailsSubmission** Web service request via SOAP takes the following form:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:lvis="lvis.webservices.smesSurveyMarkDataDelivery">
   <soapenv:Header/>
   <soapenv:Body>
      <lvis:updateMarkDetailsSubmission>
         <sessionKey>-1LPV08R3K13II|07-03-2017</sessionKey>
         <nineFigureNumber>1237812</nineFigureNumber>
         <!--Optional:-->
         <markStatus>Abandoned</markStatus>
         <!--Optional:-->
         <planNumber>AL1112</planNumber>
         <!--Optional:-->
         <groundToMarkOffset>-23.231</groundToMarkOffset>
         <!--Optional:-->
         <coverExists>No</coverExists>
         <!--Optional:-->
         <markerPostExists>No</markerPostExists>
         <!--Optional:-->
         <gnssSuitability>Moderate</gnssSuitability>
         <!--Optional:-->
         <markType>Cross Head Nail</markType>
         <comments>test</comments>
         <!--Optional:-->
         <supportingFiles>
            <!--Zero or more repetitions:-->
            <files>
               <!--Optional:-->
               <file>aGkgdGhlcmUgaWFtIHRlc3RpbmcgY3I0NTkyIHdpdGggbXVsdGlwbGUgZmlsZXM=</file>
               <!--Optional:-->
               <fileName>Ala-1-Raw</fileName>
               <!--Optional:-->
               <fileExtension>zip</fileExtension>
               <!--Optional:-->
               <fileCategory>Raw Survey Data</fileCategory>
            </files>
         </supportingFiles>
      </lvis:updateMarkDetailsSubmission>
   </soapenv:Body>
</soapenv:Envelope>
```

The response for this request is:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:updateMarkDetailsSubmissionResponse
xmlns:ns2="lvis.webservices.smesSurveyMarkDataDelivery">
         <return>
            <messages>
               <message>Survey Mark successfully submitted for approval.</message>
            </messages>
         </return>
      </ns2:updateMarkDetailsSubmissionResponse>
   </soap:Body>
</soap:Envelope>
```

If the request was successful, the user’s session will be terminated, the sessionKey will be invalidated and a message ‘Session terminated successfully.’ is returned.

Errors and exceptions

On error, a single **`messages`** container element will be returned. The same messages will be produced as per the RESTful **`updateMarkDetailsSubmission`** Web service.

12. SMES Symbology

Symbol	Description
△	**SCN Mark**. PM with adjusted GDA94 coordinates, no AHD height
+	**SCN Mark**. PCM with adjusted GDA94 coordinates, no AHD height
○	**SCN Mark**. PM with estimated GDA94 coordinates, observed AHD height
↑	**Non-SCN Mark**. PM or PCM with estimated GDA94 coordinates, no AHD height
□	**Non-SCN Mark**. PM or PCM with estimated GDA94 coordinates, estimated AHD height
△⃝	**SCN Mark**. PM with adjusted GDA94 coordinates, observed AHD height
△⃞	**SCN Mark**. PM with adjusted GDA94 coordinates, estimated AHD height
●	**Defective Mark**. PM or PCM