

mobilecommerce



LocationGateway
API Programmers Guide and Reference
Version 3.1

© 2008 Mobile Commerce Ltd. All Rights Reserved

Filename: Location Gateway API Prog Ref v3-1.doc
Author: Colin Bates
Owner: Technical Section
Status: Published

Table of Contents

Table of Contents	2
Document Version Control	2
Introduction	3
Commercial arrangements for use of the API	3
What's new in version 3?	3
Quick Start	4
LocationGateway basics	4
Calling LocationGateway from a program	7
Secure use of LocationGateway	7
Quick Start Summary	7
Guide to LocationGateway Functions	8
Invoking LocationGateway	8
Customer Authentication	8
Automatic Handset Location	8
Locate and Search	11
Manual Location and Geocoding	14
Error Handling	16
Best Practices	16
API Reference	20
Appendix A	24
Request XML document schema	24
Response XML document schema	24
Error Response XML document schema	24
Error codes	24
Search Category list	24
Search Precision Codes	24
Appendix B	25
Code example – Visual Basic	25
Code example – Java	25
Code example – C#	28

Document Version Control

V3-1 / Production Release / 7-Apr-2008
V3-0 / Production Release / 15-Jan-2007
V2-3c / Production Release / 19-Apr-2005
V2-3 / Production Release / 5-Feb-2004
V2-2 / Production Release / 13-Jun-2003
V2-1-1a / Production Release / 26-Feb-2003
V2-1 / Production Release / 20-Jan-2003

Introduction

LocationGateway is a network-level API providing a session-less method of requesting a list of merchants ordered by proximity to a defined origin point.

The API is intended to be used by a server program to obtain location-enabled data and transform it for display or use by other applications.

This Programmers Guide introduces the reader to the functions of the API and provides a detailed reference and supplementary information on how to best use the API.

Commercial arrangements for use of the API

LocationGateway is available through commercial contracts that provide for evaluation, development or production use.

Contact Bryan Stockwell, Business Development Director, for more details.

<mailto:bryan.stockwell@mobilecommerce.co.uk>

What's new in version 3?

Version 3 of the API adds an HTTPS security layer and provides support for native shapes (where provided by a network operator) in addition to standard circular locations.

Version 3.1 adds support for flexible mapping between API customers and account credentials presented to mobile network operators, to enable different types of billing arrangements.

Version 3.1 also incorporates a change in execution platform, with the LocationGateway API now operating as a dotNet application (ASP.Net) rather than an older-style ASP application. There are no functional differences resulting from this change, but the URL of the service has changed significantly.

Location shapes.

LocationGateway always returns the location of the phone as a circle centered on the co-ordinates, with a radius in metres. Some UK mobile network operators now supply location by using other shapes, notably circular arcs. LocationGateway supports returning these location shapes, where present, in addition to a circle representing the location. For more information see section "Location shapes".

Security

In this release, access to live location data can only be achieved by using HTTPS to connect to locationGateway. Location Requests using HTTP will always act in test mode and return a random location.

Additional error reporting

LocationGateway has in the past used the error number 9300 to indicate a failure to locate. In Release 3 this has been changed to provide more detailed information about the location failure, so the user can determine, for example, whether the failure was due to the caller being unreachable or the location service being unavailable.

Quick Start

This section provides basic information on using the API. Refer to the Guide section for how to use the API functions, and the Reference section for details of individual parameters.

LocationGateway basics

The LocationGateway API takes a request in the form of an HTTP POST containing an XML-formatted request document. It returns an XML-formatted response document.

The easiest way to get started with LocationGateway is to invoke it from the web-based test tool supplied at:

<http://www.mcproton.com/api/v3/testbed.asp>

To use the test tool, type or paste an XML-formatted request document into the entry window and click the search button.

You can use the test tool to develop and test XML request formats before coding them into your products.

The test tool requires a web browser with XML capability.

The minimum levels of browser we recommend are shown below

<u>Browser</u>	<u>Minimum Level</u>
Microsoft Internet Explorer	5.5 SP2
Mozilla Firefox	1.0.7
Netscape	7.2
Opera	version 9 Beta

For data security reasons, the test tool returns only random locations, and does not invoke the live mobile handset location feed function.

Locating a handset

Paste the following XML into the test tool and run it:

```
<?xml version="1.0" encoding="UTF-8"?>
<api xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=" ../api.xsd">
  <user>
    <id>5</id>
    <pwd>demo</pwd>
  </user>
  <origin>
    <msisdn>+447771123456</msisdn>
  </origin>
</api>
```

The test tool will return a formatted page showing the location of the phone (using a random location).

Merchant Search Results

		◀ Back to Test
		◀ Back to Simple Test
Search Precision	64	
Original Location	Co-ordinates	Eastings : 532053 Northings : 178202 Accuracy : 1500 Time of Location : 2006/03/15 10:03:25 GMT
Located Near	Kennington Tube Station ,London	
Postcode	SE173JF ,SE17	
Radius	1500 Metres	
Mobile Number	+447771123456	

Take a moment to review the data displayed here.

Now re-run the test tool with the same XML, but uncheck the Translate Raw XML box. This time the test tool will show the raw XML response returned by the API.

```
<?xml version="1.0" standalone="yes" ?>
- <BUSINESS_RESULTS>
  <API_VERSION>3-1</API_VERSION>
  <SEARCH_PRECISION>64</SEARCH_PRECISION>
  - <NAME>
    <![CDATA[ Barnes ]]>
  </NAME>
  - <CONTAININGNAME>
    <![CDATA[ London ]]>
  </CONTAININGNAME>
  - <POSTCODE>
    <![CDATA[ SW66NA ]]>
  </POSTCODE>
  - <CONTAININGPOSTCODE>
    <![CDATA[ SW6 ]]>
  </CONTAININGPOSTCODE>
  <MSISDN>+447771123456</MSISDN>
  <EASTING>523339</EASTING>
  <NORTHING>176775</NORTHING>
  <COORD_SYSTEM>GRID_GB</COORD_SYSTEM>
  <ZVALUE>1500</ZVALUE>
  <LOCATETIME>2006/04/07 14:11:44</LOCATETIME>
</BUSINESS_RESULTS>
```

The XML schema for the request document can be found in [Appendix A](#).

Choosing a co-ordinate system

By default, LocationGateway returns handset location using the OS National grid co-ordinate system. You can choose to receive handset location in Latitude / Longitude (GPS / WGS84) by using the <outputcoordsys> element in the <control> section. Try it by amending the previous XML as follows:

```
<api...>
  <user>
    <id>5</id>
    <pwd>demo</pwd>
  </user>
  <origin>
    <msisdn>+447771123456</msisdn>
  </origin>
  <control>
    <outputcoordsys>GPS</outputcoordsys>
  </control>
</api>
```

Using your own credentials

Contracted customers will be supplied with a set of credentials to manage their use of LocationGateway. Test your credentials now by amending the previous XML as follows:

```
<api...>
  <user>
    <id>your ID here</id>
    <pwd>your password here</pwd>
  </user>
  ...
```

If there is a problem with credentials, LocationGateway will return error code 8005.

```
<?xml version="1.0" standalone="yes" ?>
- <ERRORMSG>
  <API_VERSION>3-1</API_VERSION>
  <ERROR_CODE>8005</ERROR_CODE>
  - <ERROR_DATA>
    - <DIAGNOSTIC>
      <![CDATA[ ]]>
    </DIAGNOSTIC>
  </ERROR_DATA>
</ERRORMSG>
```

There are four permission levels that can be assigned to a user when using location requests.

- Random location only, this is where a user will be returned randomised locations for all phone numbers requested. This is the default permission level assigned to a user initially.
- WhiteListed. User can get live location feed for phone numbers on their white list only, all other numbers will be returned an error.

- Operator Limited ,this is the user only requires location from a subset of the total number of operators. They will be returned errors for devices on other operators
- Full access, the user can get location for any phone on any of the supported networks.

Other Test Tool features

The random location feature used in the test tool includes random location errors. This allows you to examine error response before designing code to handle them.

```
<?xml version="1.0" standalone="yes" ?>
- <ERRORMSG>
  <API_VERSION>3-1</API_VERSION>
  <ERROR_CODE>9300</ERROR_CODE>
- <ERROR_DATA>
  - <DIAGNOSTIC>
    <![CDATA[ Sorry we can't locate you at the moment, please try again later ]]>
  </DIAGNOSTIC>
</ERROR_DATA>
</ERRORMSG>
```

Other LocationGateway features

LocationGateway allows you to carry out a search against the Thomson Business Directory based on handset location. You can also make a search from a location of your own choice. These functions are described in more detail in the Searching section of the Reference Guide. For now, try a search by amending the previous XML to include a <Search> section as follows:

```
<api...>
<user>
<id>5</id>
<pwd>demo</pwd>
</user>
<origin>
<msisdn>+447771123456</msisdn>
</origin>
<search>
<term><id>2520</id></term>
<numresults>3</numresults>
</search>
<control>
<outputcoordsys>GPS</outputcoordsys>
</control>
</api>
```

Term ID 2520 is requesting a search for Taxis, and the results set will be limited to 3 business listings.

Merchant Search Results

[◀ Back to Test](#)
[◀ Back to Simple Test](#)

Search Precision 64
Original Location **Co-ordinates** Latitude : 51.52323
 Longitude : -0.20356
 Accuracy : 1500
 Time of Location : 2006/03/15 11:31:41 GMT

Located Near Notting Hill ,London
Postcode W105XD ,W10
Radius 1500 Metres
Mobile Number +447771123456

Name	Address	Distance	Notes
AK Car Service (Mini Cabs)	480 Harrow Rd London W93QA 02089621666 Latitude : 51.52473 Longitude : -0.20222	less than 1Km View Map	
A K Cab Service (Taxis & Private Hire)	480 Harrow Rd London W93QA 02072661566 Latitude : 51.52473 Longitude : -0.20222	less than 1Km View Map	
Windsor Cars (Mini Cabs)	309 Harrow Rd London W93RG 02072661212 Latitude : 51.52319 Longitude : -0.1979	less than 1Km View Map	

Calling LocationGateway from a program

LocationGateway can be called from within another program by an HTTP POST command to post a form containing the Request XML document, to

<http://www.mcproton.com/locationgateway/v31/find.aspx>

The Request XML must be passed through using the StrXML= parameter and must be URL-encoded. Example code can be found in Appendix B for Windows (using WinHttpRequest) and for Java (using HttpURLConnection).

Create a simple test program following this example code, and re-try the “getting started examples” shown above to get familiar with constructing requests programmatically.

Remember that LocationGateway returns random locations when used via HTTP, and does not invoke the live mobile handset location feed function.

Secure use of LocationGateway

From release 3.0 onwards, LocationGateway will return live mobile handset information only when invoked over a secure HTTPS connection. Mobile Commerce Ltd enforces encryption of all traffic over the HTTPS connection using a security certificate issued by Verisign. The secure connection to LocationGateway is

<https://www.mcproton.com/locationgateway/v31/find.aspx>

To connect to LocationGateway in secure mode you will need appropriate HTTPS libraries and your server will need to be able to accept certificates signed by Verisign. Once again, Example code can be found in Appendix B for Windows (using WinHttpRequest-secure) and for Java (using HttpURLConnection-secure). Use one of these examples to construct a secure test program, and once again re-try the “getting started” examples to validate that everything is working OK.

Quick Start Summary

You have now used LocationGateway from the test tool to become familiar with the request and response formats, and have validated that the user id and password allocated to you is active. You have used this knowledge to build a simple test program to use LocationGateway programmatically, and finally you have build a secure test program and obtained live handset location.

The next section of this manual will describe the LocationGateway function set in detail and will provide a Best Practices guide.

Guide to LocationGateway Functions

This section provides guidance and examples on using the main functions of the LocationGateway. Refer to the Reference section for full details of individual parameters.

Invoking LocationGateway

LocationGateway is invoked by using a POST command to post a form containing the request XML document. The request document is passed as the strXML parameter within the form, and the form must be URL-encoded.

Access to LocationGateway

LocationGateway operates in secure mode from <https://www.mcproton.com/locationgateway/v31/find.aspx>

but for test and development purposes it can be accessed in non-secure mode from

<http://www.mcproton.com/locationgateway/v31/find.aspx>

For live handset location, a secure HTTPS connection must be used.

Version Control

LocationGateway will be enhanced over time. Version control will be maintained by use of a standard version.release numbering scheme.

The LocationGateway version and release designators will be incorporated into the URL at directory level. The executable name will not change. This will allow the LocationGateway customer to explicitly select which version of the LocationGateway to use.

The LocationGateway version designator will be included in all outbound documents in the API_VERSION element.

The current production release of the XML LocationGateway is version 3, release1. This will be incorporated in the URL as /v31/find.aspx and in the API_VERSION element as the character value "3-1". Subsequent enhancements will be identified by incrementing the release level indicator.

The response documents may change between releases. We recommend that when moving to a new release of the LocationGateway, you feed the corresponding version of the XML schema into your XML parser.

Customer Authentication

Mobile Commerce Ltd issues a unique LocationGateway user ID and password to each customer on completion of contract formalities. This ID and password must be passed to the LocationGateway in each request document via the <user><id> and <pwd> elements.

An additional control element, <test>, provides a function for identifying non-billable transactions for system test purposes.

The <user> and <test> values associate LocationGateway usage with external billing mechanisms.

Use of live mobile phone handset location feed is only available to customers with a full commercial contract with Mobile Commerce Ltd. All other customers will be restricted to the random location feed.

Automatic Handset Location

The primary function of LocationGateway is automatic handset location. This is invoked when the user supplies a phone number using the <origin><msisdn> element in the request document.

```
<user>  
<id>your user id here</id>
```

```
<pwd>your password here</pwd>  
</user>  
<origin>  
<msisdn>447771123456</msisdn>  
</origin>
```

This will return the location of the phone, the time the location was obtained and the nearest known place name and postcode.

```
<?xml version='1.0' standalone='yes'?>  
<BUSINESS_RESULTS>  
<API_VERSION>3-0</API_VERSION>  
<SEARCH_PRECISION>64</SEARCH_PRECISION>  
<NAME><![CDATA[Elephant and Castle Rail Station]]></NAME>  
<CONTAININGNAME><![CDATA[London]]></CONTAININGNAME>  
<CONTAININGNAME2><![CDATA[London]]></CONTAININGNAME2>  
<POSTCODE><![CDATA[SE16TW  ]]></POSTCODE>  
<CONTAININGPOSTCODE><![CDATA[SE1  ]]></CONTAININGPOSTCODE>  
<MSISDN>+447771123456</MSISDN>  
<EASTING>532181</EASTING>  
<NORTHING>178873</NORTHING>  
<COORD_SYSTEM>GRID_GB</COORD_SYSTEM>  
<ZVALUE>1500</ZVALUE>  
<LOCATETIME>2006/04/07 14:36:47</LOCATETIME>  
</BUSINESS_RESULTS>
```

Co-ordinate Systems

The location of the phone is returned as geographic co-ordinates. Use the <control><outputcoordsys> to specify which co-ordinate system to be used.

```
<origin>  
<msisdn>+447771123456</msisdn>  
</origin>  
<control>  
<outputcoordsys>GRID</outputcoordsys>  
</control>
```

This will return the location of the phone as Grid co-ordinates (Eastings and Northings).

The LocationGateway supports two main types of co-ordinate systems:

- GPS
An absolute system of reference expressed as degrees of latitude and longitude values based on the equator and the Greenwich meridian respectively.
There are a number of detail variants described in specifications WGS84 and others.
- Grid reference
A relative system of reference expressed as abstract easting and northing values based on a defined grid origin position. Each nation uses its own origin position.

The LocationGateway uses the Ordnance Survey National Grid of Great Britain for its internal processes. Most automatic location systems used by mobile network operators use GPS. The LocationGateway features co-ordinate conversion between GPS and grid systems. The <control> <outputcoordsys> request element allows the user to select which co-ordinate system to return.

Northern Ireland is mapped to the Irish Grid, not the GB National Grid. The LocationGateway contains special processing to switch to Irish Grid when dealing with locations in Northern Ireland. The <COORD_SYSTEM> value in the response

message specifies which grid system is being used (GRID_GB for the GB National Grid or GRID_IRL for the Irish Grid).

Location shapes

LocationGateway always returns the location of the phone as a circle centered on the co-ordinates, with a radius in metres specified by the <zvalue> element. Expect z-values of 300-500m in dense urban areas, degrading to 1000-5000m in rural areas.

```
<BUSINESS_RESULTS>
<API_VERSION>3-0</API_VERSION>
<SEARCH_PRECISION>64</SEARCH_PRECISION>
<NAME><![CDATA[Barnes]]></NAME>
<CONTAININGNAME><![CDATA[London]]></CONTAININGNAME>
<POSTCODE><![CDATA[SW130DA]]></POSTCODE>
<CONTAININGPOSTCODE><![CDATA[SW13]]></CONTAININGPOSTCODE>
<MSISDN>447771123456</MSISDN>
<EASTING>522687</EASTING>
<NORTHING>176453</NORTHING>
<COORD_SYSTEM>GRID_GB</COORD_SYSTEM>
<ZVALUE>1500</ZVALUE>
<LOCATETIME>2004/02/04 13:18:52</LOCATETIME>
```

Some UK mobile network operators now supply more precise location by using techniques such as timing advance and sweep angles. These give rise to a variety of other shapes, notably circular arcs. LocationGateway now provides full support for shapes in addition to the standard circular presentation through optional elements in the response message.

Below is an example of a Circular_Arc Shape Element being returned.

```
<BUSINESS_RESULTS>
<API_VERSION>3-0</API_VERSION>
<SEARCH_PRECISION>64</SEARCH_PRECISION>
<NAME><![CDATA[Love Lane Industrial Estate]]></NAME>
<CONTAININGNAME><![CDATA[Cirencester]]></CONTAININGNAME>
<POSTCODE><![CDATA[GL71FA]]></POSTCODE>
<CONTAININGPOSTCODE><![CDATA[GL7]]></CONTAININGPOSTCODE>
<MSISDN>447771123456</MSISDN>
<EASTING>51.706</EASTING>
<NORTHING>-1.96472</NORTHING>
<COORD_SYSTEM>GPS</COORD_SYSTEM>
<ZVALUE>396</ZVALUE>
<LOCATETIME>2006/04/11 12:03:08</LOCATETIME>
<SHAPE>
<CIRCULAR_ARC_AREA>
<POINT>
<LL_POINT>
<LAT>51.706</LAT>
<LONG>-1.959</LONG>
</LL_POINT>
</POINT>
<IN_RADIUS>0</IN_RADIUS>
<OUT_RADIUS>791</OUT_RADIUS>
<START_ANGLE>210</START_ANGLE>
<STOP_ANGLE>330</STOP_ANGLE>
</CIRCULAR_ARC_AREA>
</SHAPE>
```

```
</BUSINESS_RESULTS>
```

When an operator supplies location using a non-circular shape, LocationGateway will convert this into a circular shape using the “largest circle” method. This method calculates the largest circle that fits entirely within the supplied shape.

Time of location

The time that the location was obtained is returned in the <LOCATETIME> element. This is supplied in a standard date/time format YYYY/MM/DD HH:MM:SS and always uses Greenwich Mean Time (GMT).

Descriptive names and post codes

The LocationGateway returns a place name in the <name>, <containingname> and <containingname2> elements to describe the location of the user’s handset. The <name> is normally a village, district or landmark, while <containingname> and <containingname2> are town and county respectively.

The LocationGateway also returns the postcode corresponding to the location of the user’s handset. The <postcode> element holds the full postcode, and the <containing postcode> holds the larger postcode area. Note that the granularity of postcodes in most urban and suburban areas is much finer than the accuracy of the location feed, so the postcode value may not be suitable for describing the location of the handset to the end-user. However, it is useful for launching most free-to-use web-based mapping services to display a map of the area around the handset.

Locate and Search

LocationGateway also provides a search function to allow the user to find information near to the handset location. This function returns a list of merchants in order of distance from a search origin point. There are three sub-types of merchant search:

Search by Category

Returns a list of merchants selected from a specified merchant classification category.

The LocationGateway request must include a valid classification ID in the <term><id> element. The LocationGateway processing will pre-validated against a table of known ID values before running a full search.

A full list of valid classification codes is included in [Appendix A](#).

Example requesting restaurant (termid 1449) nearest to postcode “GL1 4YJ”.

```
<user>
<id>5</id>
<pwd>demo</pwd>
</user>
<search>
<term>
<id>1449</id>
</term>
<numresults>2</numresults>
</search>
<origin>
<postcode>GL1 4YJ</postcode>
</origin>
```

Response gives McDonalds and The Vaults

```
<BUSINESS_RESULTS>
<API_VERSION>3-0</API_VERSION>
<SEARCH_PRECISION>16</SEARCH_PRECISION>
<NAME><![CDATA[Love Lane Industrial Estate]]></NAME>
<CONTAININGNAME><![CDATA[Cirencester]]></CONTAININGNAME>
<CONTAININGNAME2><![CDATA[Gloucestershire]]></CONTAININGNAME2>
<POSTCODE><![CDATA[GL7 1YJ]]></POSTCODE>
<CONTAININGPOSTCODE><![CDATA[GL7 1YJ]]></CONTAININGPOSTCODE>
<EASTING>402857</EASTING>
```

```

<NORTHING>200326</NORTHING>
<COORD_SYSTEM>GRID_GB</COORD_SYSTEM>
<ZVALUE/>
<LOCATETIME/>
<BUSINESS_LISTING>
<REF>TH12856406013030</REF>
<NAME><![CDATA[McDonald's Restaurants Ltd]]></NAME>
<CID>64830</CID>
<YPCLAST><![CDATA[Restaurants]]></YPCLAST>
<ADDRESS><![CDATA[Kings Meadow, Cricklade Rd]]></ADDRESS>
<DISTRICT/>
<CITY>Cirencester</CITY>
<COUNTY><![CDATA[Gloucestershire]]></COUNTY>
<POSTCODE>GL71NP</POSTCODE>
<PHONE>01285640601</PHONE>
<TOLLFREE/>
<FAX/>
<DESCR1><![CDATA[Other Restaurant]]></DESCR1>
<DESCR2/>
<DISTANCE>less than 1Km</DISTANCE>
</BUSINESS_LISTING>
<BUSINESS_LISTING>
<REF>TH12858857063030</REF>
<NAME><![CDATA[The Vaults]]></NAME>
<CID>64830</CID>
<YPCLAST><![CDATA[Restaurants]]></YPCLAST>
<ADDRESS><![CDATA[5 Beeches Rd]]></ADDRESS>
<DISTRICT/>
<CITY>Cirencester</CITY>
<COUNTY><![CDATA[Gloucestershire]]></COUNTY>
<POSTCODE>GL71BN</POSTCODE>
<PHONE>01285885706</PHONE>
<TOLLFREE/>
<FAX/>
<DESCR1><![CDATA[Other Restaurant]]></DESCR1>
<DESCR2/>
<DISTANCE>1.6 km</DISTANCE>
</BUSINESS_LISTING>
</BUSINESS_RESULTS>
    
```

Search by Merchant Name

Returns a list of merchants whose name begins with a specified text string, across all merchant classification categories.

The LocationGateway Customer must supply a text string for the merchant name in the <search><businessname> element. The LocationGateway will return merchants across all categories whose name begins with the supplied text.

Example requesting business's with a name beginning "MC" nearest to postcode "GL1 4YJ".

```

<api>
<user>
<id>5</id>
<pwd>demo</pwd>
</user>
<search>
<businessname>MC</businessname>
<numresults>2</numresults>
</search>
<origin>
<postcode>GL1 1YJ</postcode>
</origin>
</api>
    
```

This returns McDonalds Restaurant and McGills the Accountants

```

<BUSINESS_RESULTS>
<API_VERSION>3-0</API_VERSION>
<SEARCH_PRECISION>16</SEARCH_PRECISION>
<NAME><![CDATA[Love Lane Industrial Estate]]></NAME>
<CONTAININGNAME><![CDATA[Cirencester]]></CONTAININGNAME>
<CONTAININGNAME2><![CDATA[Gloucestershire]]></CONTAININGNAME2>
<POSTCODE><![CDATA[GL71YJ]]></POSTCODE>
<CONTAININGPOSTCODE><![CDATA[GL7]]></CONTAININGPOSTCODE>
<EASTING>402857</EASTING>
    
```

```

<NORTHING>200326</NORTHING>
<COORD_SYSTEM>GRID_GB</COORD_SYSTEM>
<ZVALUE/>
<LOCATETIME/>
<BUSINESS_LISTING>
<REF>TH12856406013030</REF>
<NAME><![CDATA[McDonald's Restaurants Ltd]]></NAME>
<CID>64830</CID>
<YPCLAST><![CDATA[Restaurants]]></YPCLAST>
<ADDRESS><![CDATA[Kings Meadow, Cricklade Rd]]></ADDRESS>
<DISTRICT/>
<CITY>Cirencester</CITY>
<COUNTY><![CDATA[Gloucestershire]]></COUNTY>
<POSTCODE>GL71NP</POSTCODE>
<PHONE>01285640601</PHONE>
<TOLLFREE/>
<FAX/>
<DESCR1><![CDATA[Other Restaurant]]></DESCR1>
<DESCR2/>
<DISTANCE>less than 1Km</DISTANCE>
</BUSINESS_LISTING>
<BUSINESS_LISTING>
<REF>TH12856521283030</REF>
<NAME><![CDATA[McGills Chartered Accountants]]></NAME>
<CID>1340</CID>
<YPCLAST><![CDATA[Accountants]]></YPCLAST>
<ADDRESS><![CDATA[Oakley Ho, Tetbury Rd]]></ADDRESS>
<DISTRICT/>
<CITY>Cirencester</CITY>
<COUNTY><![CDATA[Gloucestershire]]></COUNTY>
<POSTCODE>GL71US</POSTCODE>
<PHONE>01285652128</PHONE>
<TOLLFREE/>
<FAX/>
<DESCR1/>
<DESCR2/>
<DISTANCE>1.7 km</DISTANCE>
</BUSINESS_LISTING>
</BUSINESS_RESULTS>
    
```

Search by Category and Name

Returns a list of merchants whose name begins with a specified text string, from a specified merchant classification category. This is a combination of the previous two search functions. So if we search for nearest restaurants (Termid 1449) whose name begins "mc"

```

<user>
<id>5</id>
<pwd>demo</pwd>
</user>
<search>
<term>
<id>1449</id>
</term>
<businessname>MC</businessname>
<numresults>2</numresults>
</search>
<origin>
<postcode>G17 1YJ</postcode>
</origin>
    
```

We get the two nearest restaurants that start with the letters "mc".

```

<BUSINESS_RESULTS>
<API_VERSION>3-0</API_VERSION>
<SEARCH_PRECISION>16</SEARCH_PRECISION>
<NAME><![CDATA[Love Lane Industrial Estate]]></NAME>
<CONTAININGNAME><![CDATA[Cirencester]]></CONTAININGNAME>
<CONTAININGNAME2><![CDATA[Gloucestershire]]></CONTAININGNAME2>
<POSTCODE><![CDATA[GL71YJ]]></POSTCODE>
<CONTAININGPOSTCODE><![CDATA[GL71YJ]]></CONTAININGPOSTCODE>
<EASTING>402857</EASTING>
<NORTHING>200326</NORTHING>
<COORD_SYSTEM>GRID_GB</COORD_SYSTEM>
<ZVALUE/>
    
```

```

<LOCATETIME/>
<BUSINESS_LISTING>
<REF>TH12856406013030</REF>
<NAME><![CDATA[McDonald's Restaurants Ltd]]></NAME>
<CID>64830</CID>
<YPCLAST><![CDATA[Restaurants]]></YPCLAST>
<ADDRESS><![CDATA[Kings Meadow, Cricklade Rd]]></ADDRESS>
<DISTRICT/>
<CITY>Cirencester</CITY>
<COUNTY><![CDATA[Gloucestershire]]></COUNTY>
<POSTCODE>GL71NP</POSTCODE>
<PHONE>01285640601</PHONE>
<TOLLFREE/>
<FAX/>
<DESCR1><![CDATA[Other Restaurant]]></DESCR1>
<DESCR2/>
<DISTANCE>less than 1Km</DISTANCE>
</BUSINESS_LISTING>
<BUSINESS_LISTING>
<REF>TH17938786933030</REF>
<NAME><![CDATA[McDonald's Restaurants Ltd]]></NAME>
<CID>64830</CID>
<YPCLAST><![CDATA[Restaurants]]></YPCLAST>
<ADDRESS><![CDATA[West Swindon District Centre]]></ADDRESS>
<DISTRICT/>
<CITY>Swindon</CITY>
<COUNTY><![CDATA[Wiltshire]]></COUNTY>
<POSTCODE>SN57DL</POSTCODE>
<PHONE>01793878693</PHONE>
<TOLLFREE/>
<FAX/>
<DESCR1><![CDATA[Other Restaurant]]></DESCR1>
<DESCR2/>
<DISTANCE>18.3 km</DISTANCE>
</BUSINESS_LISTING>
</BUSINESS_RESULTS>
    
```

Results Set Size

The search will use a dynamic radius about the search origin to balance search efficiency with results set size. The initial proximity search will be made in a small area about the search origin. If this returns less than the requested number of records requested through element <numresults>, the search radius will be progressively doubled until the requested number of records is returned, subject to a maximum search radius of 80Km (approx 50 miles).

If element <numresults> is not supplied, the default results set size will be 10.

Manual Location and Geocoding

The LocationGateway also supports searching without use of automatic handset location. The Manual Location function sets a search origin by interpretation of placename information provided in the LocationGateway request.

The location is returned in the response document as positional co-ordinates and a placename description.

Manual location can also be used without searching for merchants, to provide a geocoding function.

Manual location is invoked when the user supplies placename information using the <origin> elements in the request document.

A geocode function is available by invoking manual location without requesting a search for nearby businesses.

Three types of self-locate are supported:

Locate by co-ordinates

This is the most precise method to locate the user. Typically this would be used where location co-ordinates are already known, for example from a GPS receiver, or from a previous LocationGateway request.

```
<origin>
<coordsys>gps</coordsys>
<lat>53.4808</lat>
<long>-2.2436</long>
</origin>
```

Locate by landmark

This is a medium precision method that can be used when the calling application is able to present the user with a menu of known landmarks such as tube stations, railway stations or public buildings.

```
<origin>
<placetype>
<type>tube</type>
<placename>Kilburn</placename>
</placetype>
</origin>
```

LocationGateway will search the list of known tube stations for any beginning with the value passed through by the placename element. If no tube stations are found, an error is returned. If more than one match is returned, the first tube station found is used. `<placename>Kilburn</placename>` will find Kilburn Tube Station, but for Kilburn Park Tube Station, use `<placename>Kilburn Park</placename>`.

Locate by address

This is a variable-precision method. LocationGateway attempts to resolve the user's location by the most precise method possible from the input provided by the user. Locate by address uses whatever combination of postcode (full or partial), town, district, street name and building number the user has been able to supply.

1. By postcode

```
<origin>
<postcode>GL7 1YJ</postcode>
</origin>
```

2. By street name and building

```
<origin>
<address2>Love Lane</address2>
<town>Cirencester</town>
</origin>
```

3. By district and town, town or town and county

```
<origin>
<district>Siddington</district>
<town>Cirencester</town>
</origin>
```

```
<origin>
<town>Newport</town>
<county>Gwent</county>
</origin>
```

The LocationGateway response includes a `<search_precision>` element which describes how precisely the LocationGateway has been able to resolve the user's position from the supplied self-location information. A list of search precision codes is found in [Appendix A](#).

Co-ordinate input

Use the `<origin><coordsys>` element in the request document to specify the co-ordinate systems type you are using for self-location, and use the `<lat>` and

<long> or <easting> and <northing> elements to supply the values. When GPS is selected, the LocationGateway supports standard decimal notation or degrees:minutes:seconds. Either notation can be signed or unsigned. For longitude <long>, degrees are measures east from the Greenwich meridian. With signed notation, a -ve value indicates degrees measured west from the meridian.

The main purpose of <coordsys> is to allow differentiation between national co-ordinate systems. Two grid reference systems are used in the UK - the Ordnance Survey National Grid for mainland Great Britain (England, Scotland & Wales), and the Irish Grid (used for Northern Ireland). The LocationGateway will accept grid co-ordinates for Northern Ireland using the Irish Grid.

Here are some examples of valid co-ordinates:

Manchester (west of Greenwich)

```
<coordsys>GPS</coordsys><long>-2.2436</long><lat>53.4808</lat>  
<coordsys>GPS</coordsys><long>357.7564</long><lat>53.4808</lat>  
<coordsys>GRID_GB</coordsys>  
<easting>383931</easting><northing>398257</northing>
```

Ipswich (East of Greenwich)

```
<coordsys>GPS</coordsys><long>1.1575</long><lat>52.0591</lat>  
<coordsys>GRID_GB</coordsys>  
<northing>244783</northing><easting>616553</easting>
```

Holywood (Northern Ireland)

```
<coordsys>GPS</coordsys><long>-5.8369</long><lat>54.6379</lat>  
<coordsys>GPS</coordsys><long>354.1631</long><lat>54.6379</lat>  
<coordsys>GRID_IRL</coordsys>  
<easting>339828</easting><northing>379173</northing>
```

Error Handling

Where the LocationGateway is unable to return any merchant data, it will return an error message using the XML error document. See [Appendix A](#) for the error response document DTD and list of error codes.

Best Practices

Define your constants

To help you evolve your program code as we add more features to the API, we recommend you set up definitions in your code for the major constants used in the API, specifically:

- Server name (currently www.mcproton.com)
- API path (currently /locationgateway/v31)
- API executable (currently find.aspx)
- API version number in response (currently 3-1)

This will facilitate changes as we introduce new API functions and versions.

Specify the network

In order to locate a phone the API needs to know the network the phone is on, so that it can request the phones location from that network. In normal usage the API will determine what network the phone is on by using the OfTel prefix tables and then by checking to see it knows that the phone has been ported to another network.

In certain cases the user may already know the network that a phone is on and may want to bypass this network look-up step. The API allows you to do this by specifying a <network> element within the <origin> element. The example below is a request forcing the API to get the location from Vodafone.

```
<api...>  
<user>
```

```
<id>5</id>
<pwd>demo</pwd>
</user>
<origin>
<msisdn>+447771123456</msisdn>
<network>UKVF</network>
</origin>
<search>
<term><id>2520</id></term>
<numresults>3</numresults>
</search>
<control>
<outputcoordsys>GPS</outputcoordsys>
</control>
</api>
```

Currently this element can have 4 values

UKO2	O2
UKOR	Orange
UKTM	T-Mobile
UKVF	Vodafone

Caching of automatic handset location

The mobile phone operators use various caching systems to hold the results of previous location requests for handsets, and may supply a cached location in response to a new request. If your application is very time-sensitive, we recommend you inspect the <LOCATETIME> value to check the age of the location response. Please note the following operator-specific points:

- On O2 UK,
Everything happens in real time all the time. When the phone is in contact with the network, all location requests are serviced immediately and directly. If the phone is not in contact with the network, O2 returns a code 720 Device Unreachable error response. This location feed does what it says on the tin - location of one handset, now.
- On Vodafone UK,
When the phone is in contact with the network, all location requests are serviced in real time and there is no evidence of caching. However, if the phone is not in contact with the network, one of two things will happen:
 - if the phone leaves the network cleanly (e.g. it has been switched off), Vodafone returns the location of the phone at the time it was switched off and gives a location date/time stamp that corresponds to the time the phone was switched off.
 - if the phone disappears from the network (e.g. if goes out of coverage or the battery is removed), Vodafone returns the last recorded location of the phone and gives a location date/time stamp that corresponds to the time that location was obtained.This means that on Vodafone, a location result with a location date/time stamp over 5 minutes old probably means that the phone cannot currently be located.
- On T-Mobile UK,
When the phone is in contact with the network, there is evidence of up to 10 minutes caching of location requests. TM has told us that they cache for around 5 minutes. There is also evidence that the location held in cache is

updated when a phone event occurs (e.g. the phone is switched on, or a call is made on the handset). This suggests to me that TM obtains location as an event-driven background task rather than a real time request, but the caching interval of 10 minutes makes it feel near-enough like real time. However, if the phone is not in contact with the network, TM returns the last recorded location of the phone and gives a location date/time stamp that corresponds to the time that location was obtained.

This means that on TM, a location result with a location date/time stamp over 15 minutes old probably means that the phone cannot currently be located.

- On Orange,
Orange have explained to us that their location feed operates entirely as an event-driven background task, with a refresh of the handset location being triggered by network events such as making or receiving a call or an SMS, moving from one cell to another or switching the phone on or off. In the event that a handset is idle and stationary (i.e. no network events occur), the location will be refreshed after 90 minutes. The location is returned along with a location date/time stamp that corresponds to the time that location was obtained. The intended outcome of these methods is that even though the system returns a location that may appear to be some while ago, the actual location is where the phone is now. That is, unless the phone is no longer in contact with the network, when one of two things will happen:
 - if the phone leaves the network cleanly (e.g. it has been switched off), Orange returns the location of the phone at the time it was switched off and gives a location date/time stamp that corresponds to the time the phone was switched off.
 - if the phone disappears from the network (e.g. if goes out of coverage or the battery is removed), Orange returns the last recorded location of the phone and gives a location date/time stamp that corresponds to the time that location was obtained.

Unfortunately this means that we have no way of telling whether an 'old' location date/time stamp means that the phone hasn't moved for n minutes, or that it has been switched off n minutes ago and may now be somewhere else.

This means that on Orange you cannot tell whether you are getting an up-to-date location or the last-known location of the handset.

You may need to allow for a few minutes difference between system clocks between your systems and those of the network operators. Mobile Commerce Ltd synchronises system clocks regularly with an accurate network time server but this is not always the case with the network operator's servers. For example we have observed that Vodafone use a bank of servers for location processing, each of which returns a slightly different time when retrieving the same last known location data !

Handling data for Northern Ireland

API release 2.3 introduced full support for locations in Northern Ireland. When results are returned using grid co-ordinate systems, a new output element <coordsys> defines whether the grid references are on the Irish Grid or the GB National Grid. The <coordsys> element is returned in the main Business Results section of the output document when a handset location has been requested. It is also returned in each Business Listing element if a business search has been requested.

We recommend that all customers who use grid co-ordinates in their own applications check the <coordsys> value to confirm the grid system in use before carrying out any geographic operations on the grid co-ordinates.

Merchant name searches

The API will look for merchants whose name begins with the character supplied in the <businessname> element. When searching for e.g. The Gap clothing stores, <businessname>Gap</businessname> will find stores entered as "Gap" but not those entered as "The Gap".

To help overcome this, we are working on standardising company name fields for the larger chains, and will be introducing new search term IDs that will search for all branches of specific chains of stores.

Street address resolution

Street names are not always unique within a City, especially London. We strongly recommend you supply additional information such as a part post-code along with the street address e.g.

```
<address2>Gloucester Road</address2>  
<postcode>SW7</postcode>.
```

Outside of London, it may be more convenient to qualify a street name by a district of a town e.g.

```
<address2>Gloucester Road</address2>  
<district>Avonmouth</district>  
<town>Bristol</town>
```

API Reference

This section describes all of the elements in the XML request documents, with formats, permitted values, defaults etc.

Name	Value	Description	Notes
<user>			Authentication and authorisation
<id>	Numeric	API customer ID as assigned by Mobile Commerce Ltd.	
<pwd>	Text	Customer password as assigned by Mobile Commerce Ltd.	
<search>			Searching for...
<term>			Classified search
<id>	Numeric	Category ID Valid merchant classification category id as defined by Mobile Commerce Ltd.	
<termname>	Text	(not supported in this release)	
<numresults>	Numeric	Number of results to return. Range is 1 – 100, default 10	
<businessname>	Text	Business Name	Max 50 chars.
<origin>			Searching from...
<msisdn>	Text	Phone number of a mobile handset	International format e.g. 447733667755
<network>	Text	The mobile network on which the handset is operating: UKVF – Vodafone UKO2 – O ₂ UKOR – Orange UKTM – T-Mobile	Optional
<coordsys>	GPS GRID_GB GRID_IRL NONE	Co-ordinate system to be used for manual input of location co-ordinates GPS = WGS84 Lat / Long GRID_GB = GB National Grid (OS) GRID_IRL = Irish National Grid NONE = No co-ordinates supplied (default)	
<lat>	Numeric	Location co-ordinates, where <coordsys> is GPS	
<long>	Numeric	Range: 0 to 360 degrees East of Greenwich or -180 West of Greenwich to +180 degrees East of Greenwich Format: Decimal (nnn.nnnnn) DMS (ddd:mm:ss.ssss)	
<easting>	Numeric	Location co-ordinates, where <coordsys> is GRID	
<northing>	Numeric		
<address1>	Numeric	Building number	
<address2>	Text	Street name	Max 150 chars
<address3>	Text	(not supported in this release)	
<district>	Text	District name	Max 30 chars.
<town>	Text	Town or City name	Max 30 chars.

<county>	Text	County	Max 30 chars.
<postcode>	Text	Postcode	Max 7 chars.
<placetype>	Text	Type of placename, for location by POI	Only TUBE supported
<placename>	Text	POI name	Name of tube station
<control>			System options
<test>	TRUE FALSE	Non-commercial transaction indicator default is FALSE	
<debug>	TRUE FALSE	(not supported in this release)	
<outputcoordsys>	GPS GRID	Co-ordinate system to be used in output stream DEFAULT is GRID	

This section describes all of the elements in the XML response documents, with formats, permitted values, defaults etc.

Name	Value	Description	Notes
<api_version>	Text	Version in format (majorversion-minorversion)	i.e. 3-0
<search_precision>	Numeric	How location was determined. Values are defined here. http://www.mcproton.com/api/v3/search_precision.xml	
<name>	Text	Name of area request was located in, normally this would be district.	
<containingname>	Text	Name of area <name> is contained in, normally this would be town.	
<containingname2>	Text	Name of area <containingname> is contained in, normally this would be county.	
<postcode>	Text	Postcode in which the request is located.	
<containingpostcode>	Text	Postcode prefix in which the request is located.	
<msisdn>	Numeric	Phone number of a mobile handset, if present in request.	International format e.g. 447733667755
<maxlfcount>	Numeric	Maximum number of Location feed requests user has.	This is only used for specific user testing.
<easting> <northing>	Numeric Numeric	Location co-ordinates of request, where <coord_system> is GRID	
<lat> <long>	Numeric Numeric	Location co-ordinates, where <coord_system> is GPS Range: 0 to 360 degrees East of Greenwich or -180 West of Greenwich to +180 degrees East of Greenwich Format: Decimal (nnn.nnnnn) DMS (ddd:mm:ss.ssss)	
<coord_system>	GPS GRID_GB GRID_IRL	Co-ordinate system to used to return coordinates GPS = WGS84 Lat / Long GRID_GB = GB National Grid (OS)	

		GRID_IRL = Irish National Grid	
<zvalue>	Numeric	Radius in metres	
<locatetime>	Text	Time at which location was resolved in format 'YYYY/MM/DD HH:MM:SS'	
<shape>		Element that defines raw shape returned by operator.	
<circular_arc_area>		Element specifying sector	
<point>		GPS	
<ll_point>		Range:	
<lat>	Numeric		
<long>	Numeric	0 to 360 degrees East of Greenwich or -180 West of Greenwich to +180 degrees East of Greenwich	
		Format:	
			Decimal (nnn.nnnnn)
			DMS (ddd:mm:ss.ssss)
<in_radius>	Numeric	The inner radius of the arc in metres	
<out_radius>	Numeric	The outer radius of the arc in metres	
<start_angle>	Numeric	Specifies the start angle in degrees clockwise from north.	
<stop_angle>	Numeric	Specifies the stop angle in degrees clockwise from north.	
<business_listing>		Defines a list of businesses that satisfy the request criteria	
<ref>	Text	Internal reference ID	
<name>	Text	Name of business	
<cid>	Numeric	Classification Id for type of business	
<ypclast>	Text	Classification name for type of business	
<address>	Text	Address line for business	
<district>	Text	District name	
<city>	Text	Town or City name	
<county>	Text	County	
<postcode>	Text	Postcode	
<phone>	Text	Contact number for business	
<tollfree>	Text	Freephone number, if available	
<hr/>			
<fax>	Numeric	Fax Number	
<descr1>	Text	Description line 1	Used to provide extra information about the business
<descr2>	Text	Description line 2	Used to provide extra information about the business
<distance>	Text	Distance from the location specified in the request	
		-	
<easting>	Numeric	Location co-ordinates of business, where	
<northing>	Numeric	<coord_system> is GRID	
<lat>	Numeric	Location co-ordinates of business, where <coord_	
<long>	Numeric	system> is GPS	
		Range:	
			0 to 360 degrees East of Greenwich

or
-180 West of Greenwich to +180
degrees East of Greenwich

Format:

Decimal (nnn.nnnnn)

DMS (ddd:mm:ss.ssss)

<coordsys>

GPS |
GRID_GB |
GRID_IRL

Co-ordinate system to used to return coordinates

Appendix A

Request XML document schema

<http://www.mcproton.com/api/v3/api.xsd>

Response XML document schema

<http://www.mcproton.com/api/v3/output.xsd>

Error Response XML document schema

<http://www.mcproton.com/api/v3/error.xsd>

Error codes

<http://www.mcproton.com/api/v3/errors.xml>

Not All Errors are included

Search Category list

http://www.mcproton.com/api/v3/terms_list.xml

This also needs to be updated as a point in time

Search Precision Codes

http://www.mcproton.com/api/v3/search_precision.xml

Appendix B

The examples below show connection to the API via HTTPS but they will also work with HTTP.

Code example – Visual Basic

This short example shows how to make an API request using Visual Basic on a Windows platform

```
Option Explicit
Private WithEvents xmlHTTP As WinHttpRequest
Private Sub Form_Load()
Dim DataToSend As String
Set xmlHTTP = New WinHttpRequest
DataToSend = "strXML=<?xml%20version=" + Chr(34) + "1.0" + Chr(34) +
"%20encoding=" & Chr(34) & "UTF-8" & Chr(34) & "?>" & _
    "<api%20xmlns:xsi=" & Chr(34) &
"http://www.w3.org/2001/XMLSchema-instance" & Chr(34) &
"%20xsi:noNamespaceSchemaLocation=" & Chr(34) & "../api.xsd" & Chr(34) & ">"
& _
    "<user><id>99</id><pwd>apples67</pwd></user>" & _
    "<search>" & _
    "<term><id>1867</id></term><numresults>10</numresults>" & _
    "<location><coordsys>GRID</coordsys></location>" & _
    "</search>" & _
    "<origin><msisdn>07771123456</msisdn></origin>" & _
    "</api>"
xmlHTTP.Open "POST", "https://www.mcproton.com/api/v3/fmn0.asp", False
xmlHTTP.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"
xmlHTTP.Send DataToSend
Set xmlHTTP = Nothing
End Sub
Private Sub xmlHTTP_OnResponseFinished()
' Occurs when the response data is complete.
Debug.Print xmlHTTP.ResponseText
End Sub
```

Code example – Java

This short example shows how to make an API request using Java. (Note if you are using HTTPS then you will need Java version 1.4.2 or above. This example references a properties file which is included below:

```
import java.net.HttpURLConnection;
import java.net.URLEncoder;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.BufferedReader;
import java.net.URL;
import java.util.Properties;
import java.io.*;

public class getLocation {
```

```
public static void main(String[] args)
{
    try
    {
        URL url;
        // Get Properties
        Properties props = new Properties();
        FileInputStream fis = new FileInputStream( "getLocation.properties" );
        props.load(fis);
        fis.close();
        // Create URL
        url = new URL(props.getProperty("URL"));
        getLocation loc = new getLocation();
        // Build Request String
        String locationRequest =
loc.LocationRequest(props.getProperty("PhoneNumber"),
                    props.getProperty("Username"),
                    props.getProperty("password"),
                    props.getProperty("Term"),
                    props.getProperty("Name"),
                    props.getProperty("Count"));

        // Get XML Response
        loc.getResponse(url,locationRequest);
    }
    catch ( Exception e )
    {
        e.printStackTrace() ;
    }
}

public String getResponse (URL _url, String _LocationRequest)
{
    HttpURLConnection c = null;
    InputStream is = null;
    OutputStream os = null;
    StringBuffer b = new StringBuffer( );
    try
    {
        c = (HttpURLConnection)_url.openConnection();
        c.setRequestProperty("Content-Type", "application/x-www-form-
urlencoded");
        c.setDoOutput(true);
        c.setDoInput(true);
        os = c.getOutputStream();
        // Send HTTP request
        os.write(_LocationRequest.getBytes());
        // Get HTTP response
        is = c.getInputStream( );

        int ch;
        while ((ch = is.read( )) != -1)
        {
            b.append((char) ch);
        }
    }
}
```

```
    }
  }
  catch ( Exception e)
  {
    e.printStackTrace();
  }
  finally
  {
    System.out.println(b.toString());
    System.out.flush();
    return(b.toString());
  }
}

private String LocationRequest(String PhoneNumber, String Username, String
password,
                               String term, String Name, String count) throws
java.io.UnsupportedEncodingException
{
  // If term is present in properties file then use this otherwise use the
business name
  // from the properties file
  String searchtype;
  if (term != null )
  {
    searchtype = "<term><id>"+term+"</id></term>";
  }
  else
  {
    searchtype = "<businessname>"+Name+"</businessname>";
  }
  String strXML = "<?xml version=\\"1.0\\" encoding=\\"UTF-8\\"?>" +
    "<api xmlns:xsi=\\"http://www.w3.org/2001/XMLSchema-
instance\\" xsi:noNamespaceSchemaLocation=\\"../api.xsd\\">" +
    "<user><id>" + Username + "</id><pwd>"+
password+"</pwd></user>" +
    "<search>" + searchtype +
"<numresults>"+count+"</numresults></search>" +
"<origin><msisdn>"+PhoneNumber+"</msisdn><coordsys>GPS</coordsys></origin>"
+
"<control><outputcoordsys>GPS</outputcoordsys></control>" +
    "</api>";
  return "strXML="+URLEncoder.encode(strXML,"UTF-8") ;
}
}
```

Here is the getlocation.properties file referred to in the above example

```
URL=https://www.mcproton.com/api/v3/fmm0.asp
PhoneNumber=447712877412
Username=99
password=apples67
Term=1867
Count=1
```

Code example – C#

The following is a command line application which uses the LocationGateway.

```
using System;
using System.Net;
using System.IO;

/// <summary>
/// Summary description for Class1.
/// </summary>
class ConsoleAPIApp
{
    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    static void Main(string[] args)
    {
        string XmlData =
            "strXML=<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
            "<api xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" " +
            "xsi:noNamespaceSchemaLocation=\"../api.xsd\">" +
            "<user><id>99</id><pwd>apples67</pwd></user>" +
            "<origin><msisdn>447123456789</msisdn></origin>" +
            "<control><outputcoordsys>GRID</outputcoordsys></control></api>";

        // Create a request using a URL that can receive a post.
        WebRequest request = WebRequest.Create
            ("https://www.mcproton.com/api/v3/fmn0.asp");

        XmlData=XmlData.Replace(" ", "+");
        // Set the Method property of the request to POST.
        request.Method = "POST";
        // Create POST data and convert it to a byte array.

        byte[] byteArray = System.Text.Encoding.UTF8.GetBytes(XmlData);
        // Set the ContentType property of the WebRequest.
        request.ContentType = "application/x-www-form-urlencoded";
        // Set the ContentLength property of the WebRequest.
        request.ContentLength = byteArray.Length;
        // Get the request stream.
        Stream dataStream = request.GetRequestStream();
        // Write the data to the request stream.
        dataStream.Write (byteArray, 0, byteArray.Length);
        // Close the Stream object.
        dataStream.Close ();
        // Get the response.
        WebResponse response = request.GetResponse ();
        // Get the stream containing content returned by the server.
        dataStream = response.GetResponseStream ();
        // Open the stream using a StreamReader for easy access.
        StreamReader reader = new StreamReader (dataStream);
        // Read the content.
        string responseFromServer = reader.ReadToEnd ();
        // Display the content.

        // Clean up the streams.
        reader.Close ();
        dataStream.Close ();
        response.Close ();
        System.Console.WriteLine(responseFromServer);
    }
}
```

END OF DOCUMENT