$\mathcal{MC}^2$

The XML API

Rev. 3.0.0

Wayfinder Systems AB

Contents

This copy (Rev. 3.0.0) is generated September 7, 2010, 9:09.

External API — XML

This is a description of the XML interface to the $\mathcal{MC}^2$-system developed at Wayfinder Systems AB. In addition to the formal definition of the Application Programming Interface (API), a summary and some examples of the usage are also included in this document. It is assumed that the reader has some knowledge of XML. More information about XML can be found at the World Wide Web Consortium (W3C) web site at http://www.w3.org/XML/.

1 Summary

The main purpose of this interface is to provide a general API to the $\mathcal{MC}^2$-system. The API can be used by a third party to incorporate our services into their system. The main service for the $\mathcal{MC}^2$-system is to provide an answer to the question "How do I best go from A to B?". The origin and destination can for example be given to the system in terms of a coordinate, a street address, or a company name. Multiple destinations will result in a route from the origin to the closest destination.

The following services are available:

Search By sending a text search string into the $\mathcal{MC}^2$-system the matching items are returned to the caller. These items could then be used as origins or destinations. If the answer contains a category, this can be expanded to its content. Some optional parameters can be provided to make it possible for the user to set the search criterion.
Example: By sending a request containing "`Lund, Bar`" the answer might contain the street item "`Baravägen`" and the company item "`Bara Elektronik AB`", both located in the city of Lund.

Route It is possible to send an origin and a destination in a route request to the $\mathcal{MC}^2$-system and get the best route between them in reply. The origin and destinations can be specified either as coordinates, items from a previous search requests, or an entire category of companies or objects. Some optional parameters can be provided to adjust the choice of the optimal route.
Example 1: A request that contains the coordinate `(55.718, 13.190)` as origin and the point of interest named "`Lund, Station`" as destination, will get a reply containing a route from the street address that is closest to the given coordinate, (Baravägen 1), to the Central Train Station of Lund.
Example 2: If the request contains the company item "`Lund, Wayfinder Systems`" as origin, and both the restaurant "`Lund, Pizzeria Portofino`" and the restaurant "`Lund, Pizzeria Fäladstorget`" as destinations, the reply will contain a route from Wayfinder Systems to the closest one of the two pizzerias.
Example 3: A request, that contains the company item "`Lund, Wayfinder Systems`" as origin and the company category item "`Restaurant`" as destination, will be replied to with a route to the restaurant that is closest to the address of Wayfinder Systems.

Expand Items, including categories of companies, that can be returned as a reply to a Search request may be expanded. By expanding a category we mean that the items in that category will be returned.
Example: Sending an expand request for the area item "`Lund`" and the category item "`Restaurant`", will result in a reply containing a list of all restaurants in Lund.

User handling It is possible to update the user profiles by sending a user request to the $\mathcal{MC}^2$-system. The user profiles contains information about the user's vehicle, phone and other preferences that the user might have. User handling functions also include user login, verification of sessions, user logout, logging and debiting.

Positioning It is possible to ask for the approximate position of a certain mobile phone. The availability of this feature is currently limited by network hardware availibility, and user privacy considerations.

SMS Route descriptions or other text may be formatted for display on an SMS capable phone, and the SMSes may be sent to the mobile phone of a user.

E-mail Route descriptions or other text may be formatted and sent as e-mails.

Traffic The caller can get graphic traffic information for a specified area.

2 Sending Request to a Server

2.1 Connecting to a Server

The connection to the server is HTTP over TCP/IP. HTTP Keep-Alive is supported.
Normally the communication is encrypted using SSL. If it is not possible to use SSL, you may use insecure communication without SSL.

2.2 Request Encapsulation

The request is formatted as an XML document that is sent as the body of an HTTP POST request. `Content-Length` is required and must be the length of the XML document. The `Content-Type` should be `text/xml` and the URI should be `/xmlfile`. Either is sufficient for the server to treat it as an XML request.
Example of a HTTP encapsulated XML request document:

```
POST /xmlfile HTTP/1.0
Content-Type: text/xml
Content-Length: 442

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE isab-mc2>

<isab-mc2>
   <auth>
      <auth_user>username</auth_user>
      <auth_passwd>password</auth_passwd>
   </auth>

   <user_login_request transaction_id="Login1"
                       user_create_session="true">
      <user_name>ausername</user_name>
      <user_password>apassword</user_password>
      <user_service>HTML</user_service>
   </user_login_request>

</isab-mc2>
```

2.3 Additional server communication features

The server also supports content encoding. This is used to compress the body of the reply sent from the server to the client. The supported compressions are “gzip”, alias “x-gzip”, and “deflate”.

To enable this feature the client should add an http header field “Accept-Encoding” with the value of the compression wanted. A http header might look like this.

```
POST /xmlfile HTTP/1.0
Content-Type: text/xml
Accept-Encoding: gzip
Content-Length: 442
```

The server then replies with “Content-Encoding” http header field with the name of the encoding, compression, used on the body.

```
HTTP/1.0 200 OK
Content-Encoding: gzip
Content-Length: 1046
Content-Type: text/xml
```

The body of such an reply must be decoded, uncompressed, before it can be used.

Please note that even if the client says it accepts a supported encoding the server might choose not to encode the reply body. For example if the compressed body would become larger than the uncompressed one.

The client must check if the reply contains a content-coding http header field before applying any decoding.

2.3.1 Pre agreed encoding

If the request is sent to “xsdata” then the XML request document is presumed to be gziped and then encoded using our encryption. To make a “xsdata” request the XML document is first gziped, it can be left uncompressed, and then enoded, and this is not optional, our predefined encryption. The server doesn’t check nor use any encoding header and it sends the content type “application/binary”. The reply from the server must first be unencrypted, and this is not optional, using our predefined encryption. Then checked if it is gziped and gunziped if so.

3 Request

A detailed description of each request supported by the $\mathcal{MC}^2$-system can be found in the section with the same name as the request, e.g. *route_request* can be found in the route section, section 11.1.

3.1 Auth

Each request sent to the $\mathcal{MC}^2$-system must contain an auth element.

```
<!ELEMENT auth ( (auth_user, auth_passwd, user_service?) |
                 (user_session_id, user_session_key, user_service) |
                 auth_activate_request |
                 (uin, auth_token) )>
<!ATTLIST auth indentingandlinebreaks %bool; "true"
               development %bool; "false"
               client_type CDATA #IMPLIED
```

```
                client_lang %language_t; #IMPLIED >
<!ELEMENT auth_user (#PCDATA)>
<!ELEMENT auth_passwd (#PCDATA)>
<!ENTITY % user_service_t "(WAP|HTML)">
<!ELEMENT user_service (#PCDATA)>
<!ELEMENT auth_activate_request EMPTY>
<!ELEMENT uin (#PCDATA)>
```

Name	Type	Description
development	boolean	Use verbose logging
client_type	string	Type of client the request comes from
client_lang	string	The general language of the client
indentingandlinebreaks	boolean	indicates whether ignorable whitespace should be added to the reply to make it more human readable. If this attribute is false then the reply becomes smaller, and faster to parse for an automatic XML parser.
user_service	string	Describes the service that the request should be authenticated for. See *user_service_t*
auth_activate_request		Used when one doesn't have a user but whants to create one using *activate_request*

The authorization part of the document. This part contains access information about the one sending the XML-document. There are three main ways to authenticate a user. The first one is to use a user name, *auth_user* and a password, *auth_passwd*, and the second one is to verify an active session from a previous *user_login_request*. The third way is to use an *uin* and an *auth_token* from an *activate_request*.

4 Reply

For each request (see section 3 – *Request*) that the $\mathcal{MC}^2$ system processes, a reply is sent back to the caller. A detailed description of each reply can be found in the section with the same name as the reply, e.g. *route_reply* is in the route section, section 11.2.

If a general problem occurs, e.g. an invalid user name or password, the reply contains only a status code and a status message describing the problem. Optionally, a status URI may be added as well.

5 Common Entities

```
<!ENTITY % number  "NMTOKEN">      <!-- a number, format [0-9]+ -->
```

A number.

```
<!ENTITY % bool    "(true|false)"> <!-- true or false -->
```

Boolean "true" or "false".

```
<!ENTITY % vdata   "CDATA">        <!-- attribute value -->
```

Common type for attributes.

```
<!ENTITY % HREF    "%vdata;">      <!-- URI, URL or URN designating a
                                       hypertext node. -->
```

An URI, URL or URN.

```
<!ENTITY % size_t  "%number;">     <!-- a number but inf is allowed -->
```

A size, may be special `inf`, infinity, value.

```
<!ENTITY % time_t  "CDATA">        <!-- Time since the Epoch
                                       (00:00:00 UTC, January 1, 1970),
                                       measured in seconds. -->
```

Time since the Epoch (00:00:00 UTC, January 1, 1970), measured in seconds.

```
<!ENTITY % coordinate_t "CDATA">    <!-- A Latitude or longitude -->
```

A coordinate, see *position_system_t* for the possible coordinate formats.

```
<!-- position_system_t WGS84(GPS) -->
<!ENTITY % position_system_t "(WGS84|MC2|WGS84Rad|WGS84Deg)">
```

The supported coordinate systems and formats.

The WGS84 format is: (N|S|E|W) D(D*)° MM' SS[.dddd]", as used in the coordinate "*N 69° 03' 35.7840", E 24° 09' 58.8238""*. That is, first one letter for point of compass, from (N|S|E|W). *N* and *S* are used for latitudes. *E* and *W* are used for longitudes. Then D(D*) is the number of degrees, with one or more digits. *D* represents a digit in degrees. Then MM, the number of minutes, with exactly two digits. Pad with leading zeroes to make this two digits. *M* represents a digit in minutes. Then SS, the number of seconds, with exactly two digits. Pad with leading zeroes to make this two digits. *S* represents a digit in seconds. Then an optional 4 digit decimal number [.dddd], the number of milliseconds. *d* represents a digit in milliseconds.

An example of a boundingbox in the *WGS84* format, enclosing an area in Spain, is

```
<boundingbox position_sytem="WGS84"
             north_lat="N 415633.277"
             west_lon="E 12408.425"
             south_lat="N 405516.471"
             east_lon="E 25436.657" />
```

The MC2 format is a number, digits only, as in the coordinate "*664731631, 157347616*". Use of the MC2 format is recommended, unless you want to present it to an end user.

An example of a boundingbox in the *MC2* format, enclosing the western part of Baravägen in Lund, Sweden.

```
<boundingbox position_sytem="MC2"
             north_lat="664759427"
             west_lon="157299266"
             south_lat="664705227"
             east_lon="157395490" />
```

The WGS84Rad format is a radian angle using the WGS84 coordinate system.

An example of a boundingbox in the *WGS84Rad* format, enclosing the western part of Baravägen in Lund, Sweden.

```
<boundingbox position_sytem="WGS84Rad"
             north_lat="0.9724885841"
             west_lon="0.2301159401"
             south_lat="0.9724092940"
             east_lon="0.2302567079" />
```

The WGS84Deg format is a degree angle using the WGS84 coordinate system.

An example of a boundingbox in the *WGS84Deg* format, enclosing the western part of Baravägen in Lund, Sweden.

```
<boundingbox position_sytem="WGS84Deg"
             north_lat="55.71949149"
             west_lon="13.18467217"
             south_lat="55.71494850"
             east_lon="13.19273757" />
```

```
<!ENTITY % image_display_type "(std|wap)">
```

The type of display that an image is designated for, that is, a standard device or a more limited WAP device.

```
<!ENTITY % route_image_format_t "(png|gif|wbmp)">
```

The types of image formats. The formats *png* and *gif* may result in large image sizes, whereas the *wbmp* is limited to black and white. Additional image format types may be supported in future versions of the API.

```
<!ENTITY % search_item_type_t "(street|pointofinterest|category|misc|
                                other)">
```

The types of *search_item*. Additional search item types may be supported in future versions of the API.

```
<!ENTITY % message_t "(html|wml|smil)">
```

The types of content of messages that can be sent. Additional message types may be supported in future versions of the API.

```
<!ENTITY % route_turn_image_t "(map|pictogram|pictogram_set_1|
                                pictogram_set_2|pictogram_set_3|
                                pictogram_set_4|pictogram_set_5)">
```

The types of route turn images. Either graphical maps or symbolic pictograms. Additional route turn image types may be supported in future versions of the API.

```
<!ENTITY % sort_distance_t "(radius|route)">
```

The types of distances to sort by, either radius, as the crow fly, distance or route, the driving, distance.

```
<!ENTITY % route_cost_t "(distance|time|time_with_disturbances)">
```

The types of route costs to optimize by. The possible values are:

Value	Description
distance	The least number of meters.
time	The shortest time.
time_with_disturbances	The shortest time, taking account traffic disturbances.

```
<!ENTITY % road_side_t "(left_side|right_side|unknown_side|undefined_side|
                         side_does_not_matter|left_side_exit|
                         right_side_exit)">
```

The possible sides of a street. The *undefined_side*, *side_does_not_matter*, *left_side_exit* and *right_side_exit* is only used to define a starting point of a route.

Value	Description
left_side	To the left of the road.
right_side	To the right of the road.
unknown_side	Unknown side of street.
undefined_side	Undefined side of street.
side_does_not_matter	Both sides are equally easy to start.
left_side_exit	Car is driving out of an exit on the left side
right_side_exit	Car is driving out of an exit on the right side.

```
<!ENTITY % landmarklocation_t "(after|before|in|at|pass|into|arrive|
                                undefinedlocation)">
```

The location of the turn relative to the landmark.

Value	Description
after	The turn is after the landmark.
before	The turn is before the landmark.
in	The turn is in the landmark.
at	The turn is at the landmark.
pass	The landmark has to be passed before the landmark.
into	The turn is in the landmark. The landmark could be e.g. a built-up area or a country.
arrive	The turn is where you arrive at the landmark (from e.g. a ferry).
undefinedlocation	The location is unknown.

```
<!ENTITY % landmark_t "(builtUpArea|railway|area|poi|signPost|country|
                        countryAndBuiltUpArea|passedStreet|accident|
                        roadwork|camera|speedTrap|police|weather|
                        trafficGen|blackspot|userDefinedCamera)">
```

The types of landmarks:

Value	Description
builtUpArea	A Built Up Area.
railway	A railway.
area	An area, like park.
poi	A Point Of Interest.
signPost	A sign post.
country	A country.
countryAndBuiltUpArea	A country and build up area.
passedStreet	Street passed before the turn.
accident	Traffic accident on route.
roadwork	Traffic, roadwork on route.
camera	Traffic, speed camera on route.
speedTrap	Traffic, speed camera on route.
police	Traffic, police activity.
weather	Traffic, weather contitions.
trafficGen	Traffic, undefined.
blackspot	Traffic, blackspot.
userDefinedCamera	Traffic, speed camera on route reported by user.

```
<!ENTITY % top_region_t "(country|state|internationalRegion|metaregion)">
```

The types of top regions:

Value	Description
country	A country, Sweden.
state	A state, Kentucky.
internationalRegion	An international region, Medicon Valley.
metaregion	A group of other regions like Europe or Scandinavia. Can not be used in searches only in region access.

```
<!ENTITY % language_t "(swedish|english|german|danish|italian|dutch|
                        spanish|french|welch|finnish|norwegian|portuguese|
                        czech|hungarian|polish|greek|american|albanian|
                        basque|catalan|frisian|irish|galician|
                        letzeburgesch|raetoRomance|serboCroatian|slovenian|
                        valencian|slovak|russian|turkish|arabic|
                        eng|swe|ger|dan|ita|dut|spa|fre|wel|fin|nor|por|
                        eng_usa|cze|alb|baq|cat|fry|gle|glg|ltz|roh|scr|
                        slv|hun|gre|pol|slo|rus|tur|ara|chi|est|lav|lit
                        tha|bul|ind|may|isl|jpa|amh|hye|tgl|bel|ben|mya|
                        hrv|fas|gla|kat|guj|heb|hin|kan|kaz|khm|kor|lao|
                        mkd|mal|mar|mol|mon|pan|ron|srp|sin|som|swa|tam|
                        tel|bod|tir|tuk|ukr|urd|vie|zul|sot|bun|bos|sla|
                        bet|mat|scc|ukl|mlt|zh-hant|zh-hant-hk)">
```

The diffenent languages. Swedish, english, german, danish, italian, dutch, spanish, french, finnish, norwegian, portuguese, czech, hungarian, polish, greek, american, slovak, russia, slovenian, turkish, chinese and traditional chinese is supported in output such as route descriptions. There is also the possibility to use iso639-3 code with optional country dialect in ISO 3166-1 alpha-3.

```
<!ENTITY % route_vehicle_t "(passengercar|pedestrian|taxi)">
```

Type of transportation:

Value	Description
passengercar	Private passenger car
pedestrian	Pedestrian
taxi	Taxi

Additional routing vehicle types may be supported in future version of the API.

```
<!ENTITY % matchtype_t "(close|full|exact|
                         closefull|wildcard|allwords|
                         phonetic|editdistance)">
```

The types of string matching methods. This controls the way that the user input string, s_u, is compared with the database string, s_{db}.

Value	Description
close	Match if s_u is a substring of s_{db}, when spaces, dashes, and otherwise non alphanumeric characters are disregarded. This is the recommended matchtype.
exact	Match if s_u is a substring of s_{db}.
full	Match if s_u is same as s_{db}. Same as *exact*, but strings must also be the same length.
closefull	Same as *close*, but strings must also be the same length.
wildcard	Not implemented, yet. User input string may include wildcards, similar to "*" and "?".
allwords	All words in the user input string must be present in the database string. The order of the words does not matter.
phonetic	This selection allows for matching strings with similar phonetics.
editdistance	This matches strings allowing for a few misspellings, by the user and the database.

```
<!ENTITY % wordmatch_t "(beginning|anywhere|wildcardpart|beginningofword)">
```

The types of word matching:

Value	Description
beginning	The user input string must match the beginning of the database string.
anywhere	The user input string may be found anywhere in the database string.
wildcardpart	This should be used together with the "wildcard" *string-matching_t*.
beginningofword	The user input string is matched to the beginning of any word of the name in the database.

```
<!ENTITY % sorttype_t "(no_sort|alfa_sort|confidence_sort)">
```

The types of sorting:

Value	Description
no_sort	The result will not be sorted.
alfa_sort	Matches are sorted in alphabetical order.
confidence_sort	Matches are sorted by confidence, attempting to place the most probable matches on top of the match list.

```
<!ENTITY % transactionBased_t "(no_transactions|transactions|
                                transaction_days)">
```

The types of transaction types for a user:

Value	Description
no_transactions	Transactions not used for user.
transactions	Transactions per request is used for user.
transaction_days	Transactions per 24h day is used for user.

```
<!ENTITY % user_service_t "(ROUTE)">
<!ENTITY % user_method_t "(WAP|HTML|NAV|XML|SMS|OPERATOR)">
```

The types and methods of services.

```
<!ENTITY % poi_info_t "(dont_show|text|url|wap_url|email|phone_number|
                        mobile_phone|fax_number|contact_info|short_info|
                        vis_address|vis_house_nbr|vis_zip_code|
                        vis_complete_zip|vis_zip_area|vis_full_address|
                        brandname|short_description|long_description|
                        citypart|state|neighborhood|open_hours|
                        nearest_train|start_date|end_date|start_time|
                        end_time|accommodation_type|check_in|check_out|
                        nbr_of_rooms|single_room_from|double_room_from|
                        triple_room_from|suite_from|extra_bed_from|
                        weekend_rate|nonhotel_cost|breakfast|
                        hotel_services|credit_card|special_feature|
                        conferences|average_cost|booking_advisable|
                        admission_charge|home_delivery|disabled_access|
                        takeaway_available|allowed_to_bring_alcohol|
                        type_food|decor|image_url|supplier|owner|
                        price_petrol_superplus|price_petrol_super|
                        price_petrol_normal|price_diesel|
                        price_biodiesel|free_of_charge|
                        tracking_data|post_address|post_zip_area|
                        post_zip_code|open_for_season|
                        ski_mountain_min_max_height|
                        snow_depth_valley_mountain|
                        snow_quality|lifts_open_total|ski_slopes_open_total|
                        cross_country_skiing_km|glacier_area|last_snowfall|
                        booking_url|booking_phone_number|
                        special_flag)" >
```

The types of poi information fields.

The special_flag field means that if you don’t find any special field you know how to handle, this favorite should not be shown. Special fields are like tracking_data which requires a tracking request to get the latest coordinate for the tracked user.

6 Common Elements

```
<!ELEMENT status_code (#PCDATA)>
```

The status code of the transaction.

Code	Description
0	The request succeeded.
-1	The general error code. There was a problem with the request.
-2	The request was malformed, see *status_message* for detailed error.
-3	The request timed out. There was an internal timeout while processing the request.
-4	Outside map coverage. The request was for an area outside the map coverage of the server.
-5	Outside allowed area. The request was for an area outside the map coverage the user is allowed to use.

Each request may define additional status codes.

The status codes that may be returned from an user authentication request. The *auth* element, *user_login_request*, *user_verify_request*, *user_show_request* and *user_find_request* uses these error codes.

Code	Description
-201	Access denied. The user does not have access to the requested service or data.
-202	Unknown user. The user does not exist.
-203	Invalid login. The login and password does not match.
-204	Invalid session. The session is not valid.
-205	Session has expired, login again. The session has been unused for too long, login again to get a new one.
-206	Expired user. The user no longer has access to the service.
-207	Unknown token. The token does not match. Use *activate_request* to get new.
-208	Expired token. The token is too old. Use the new *auth_token* in this reply.
-209	Insufficient credit. Silver user using Gold client.
-210	Important data in reply like new server auth bob and Server list. If Server list sent with this code it is not needed to redirect immediately just store the new server list and/or auth bob.
-211	Redirect. See Server list in reply for new server to use. There may be a server auth bob in the reply too.
-212	Not on backup server. You can’t do that on this backup server.
-213	We can’t create an account for you. Try activation code.
-214	Version lock. The user is not allowed to use the current client software.
-215	License key owned by more than one user. The hardware_key in the request is owned by several users and we cannot determine which to use.
-401	External auth client not from the external entity it should be. Check installed client application, SIM and access point.
-402	External auth client is authenticated but external entity says that the user hasn’t access. Buy some extension.

```
<!ELEMENT status_message (#PCDATA)>
```

A text message describing the status of the transaction. This may be, e.g., “Ok”, or “Access Denied”.

```
<!ELEMENT status_uri EMPTY>
<!ATTLIST status_uri href %HREF; #REQUIRED >
```

An optional URI that can be used to present the error to the user.

```
<!ELEMENT status_code_extended (#PCDATA)>
```

A extended status code that can be sent to Content Window.

```
<!ELEMENT name ( #PCDATA )>
```

A name of a *search_item* or a *search_area*, e.g., “Lund”.

```
<!ELEMENT search_item ( name, itemid, streetnbr?, explicit_itemid?,
                       location_name?, lat?, lon?, category_list?
                       boundingbox?
                       search_area*, info_item?, info_field* )>
```

The element that represents an item of the *search_item_type_t* types. This type of item is sent by the server in reply to some requests, such as *search_request*. The items may also be used by the client in subsequent requests. The contained *search_area*s is *search_area*s that this *search_item* is located in.

Name	Type	Description
name	string	Name of the *search_item*
itemid	string	A unique id for this item
streetnbr	integer	The streets number
explicit_itemid		Obsolete, do not use.
location_name	string	Name of the location.
lat		Latitude coordinate.
lon		Longitude coordinate.
category_list		The categories the item belongs to.
boundingbox		Boundingbox.
search_area		See *search_area*
info_item		Added if *include_info_item* is *true*
info_field		Added if *include_info_fields* is *true*

```
<!ATTLIST search_item search_item_type %search_item_type_t; #REQUIRED
                      image CDATA #IMPLIED
                      category_image CDATA #IMPLIED
                      provider_image CDATA #IMPLIED
                      brand_image CDATA #IMPLIED
                      advert %bool; #IMPLIED >
```

Name	Type	Description
image	string	Image name wihtout file extension.
advert	boolean	Is set to true if the item is an advertisement. Default false.

```
<!ELEMENT itemid ( #PCDATA )>
```

The identifier of a *search_item*.

```
<!ELEMENT streetnbr ( #PCDATA )>
```

The street number of a search_item on the street. If a *street_item* is located on the address “14 Grosvenor Crescent”, its *streetnbr* would be “14”.

```
<!ELEMENT explicit_itemid ( #PCDATA )>
```

Obsolete, exists only for backward compatibility with old clients. Do not use this. Please use coordinates (latitude, longitude) instead.

```
<!ELEMENT location_name ( #PCDATA )>
```

It contains the name of the location of the *search_item*.

```
<!ENTITY % search_area_type_t "(municipal|city|citypart|zipcode|ziparea|
                                country|other)">
```

The types of *search_area*.

Value	Description
municipal	An administrative area.
city	A city, all sizes from small rural to a mega city.
citypart	A small part of a city.
zipcode	A postal area identified by id.
ziparea	A postal area identified by name.
country	Country name.

Additional types of areas may be supported in future versions of the API.

```
<!ELEMENT search_area ( name, areaid, location_name?,
                        lat?, lon?, boundingbox?, top_region_id?,
                        search_area* )>
```

The element that represents an area of the *search_area_type_t* types. The lat and lon is a center point of the *search_area* not necessarily the geographic center but a center like town square. The contained *search_area*s is *search_area*s that this *search_area* is part of. The *location_name* is a string with the name of the location of the *search_area*.

Name	Type	Description
name	string	Name of the area.
areaid	string	Unique area id.
location_name	string	Name of the location of the area.
boundingbox		The areas boundingbox.
lat lon		The latitude and longitude center point of the area, not necessarily the geographic center but a center like town square.
top_region_id	integer	The top region in which the area resides.
search_area		A larger area for which this area belong to.

<!ATTLIST search_area search_area_type %search_area_type_t; #REQUIRED>

The type of search_area.

<!ELEMENT areaid (#PCDATA)>

The identifier of a search_area.

<!ELEMENT position_item (lat, lon, angle?)>

Item describing a position. This can for example be used as an origin or a destination when routing. In that case the coordinates are translated to the nearest point on a street.

Name	Type	Description
lat		Latitude.
lon		Longitude.
angle		The angle is an optional part of a position. It is used to help finding the best streat near the position. The angle is clockwise from 0 to 360 where 0 and 360 is north.

<!ATTLIST position_item position_system %position_system_t; #REQUIRED>

The type of positioning reference system used in a *position_item*.

```
<!ELEMENT lat ( #PCDATA )>
```

The latitude of a position or *position_item*.

```
<!ELEMENT lon ( #PCDATA )>
```

The longitude of a position or *position_item*.

```
<!ELEMENT angle ( #PCDATA ) >
```

The angle is an optional part of a position. It is used to help finding the best streat near the position. The angle is clockwise from 0 to 360 where 0 and 360 is north.

```
<!ELEMENT boundingbox EMPTY>
<!ATTLIST boundingbox
            position_sytem %position_system_t; #REQUIRED
            north_lat CDATA #REQUIRED
            west_lon  CDATA #REQUIRED
            south_lat CDATA #REQUIRED
            east_lon  CDATA #REQUIRED >
```

Name	Type	Description
position_sytem		Which position system the latitude and longitude are expressed in.
nort_lat		North latitude.
west_lon		West longitude.
south_lat		South latitude.
east_lon		East longitude.

A bounding box with a defined positioning reference system. See figure 1. A *boundingbox* can be imagined as a form of rectangle that circumscribes another object, such as a route or a section of a map.

```
<!ELEMENT image_settings EMPTY>
<!ATTLIST image_settings
            image_show_street_main %bool; "true"
            image_show_street_first %bool; "true"
            image_show_street_second %bool; "true"
            image_show_street_third %bool; "true"
            image_show_street_fourth %bool; "true"
            image_show_builtup_area %bool; "true"
            image_show_park %bool; "true"
```

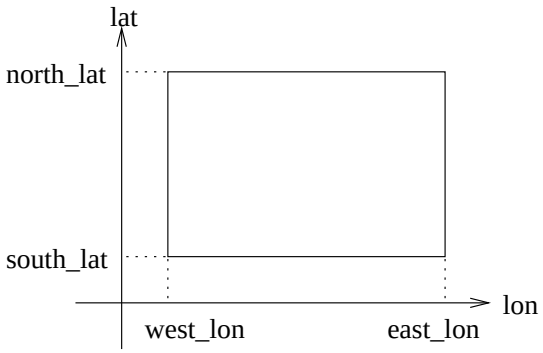

Figure 1: A boundingbox.

```
image_show_forest %bool; "true"
image_show_building %bool; "true"
image_show_water %bool; "true"
image_show_island %bool; "true"
image_show_pedestrianarea %bool; "true"
image_show_aircraftroad %bool; "true"
image_show_land %bool; "true" >
```

Set what to show on images.

Name	Type	Description
image_show_street_main	boolean	Include main roads in the image
image_show_street_first	boolean	Include level 1 roads. These are smaller than main roads.
image_show_street_second	boolean	Include level 2 roads. These are smaller than level 1 roads.
image_show_street_third	boolean	Include level 3 roads. These are smaller than level 2 roads.
image_show_street_fourth	boolean	Include level 4 roads. These are smaller than level 3 roads.
image_show_builtup_area	boolean	Include built up area polygons.
image_show_park	boolean	Include parks.
image_show_forest	boolean	Include forests.
image_show_building	boolean	Include areas of larger buildings, such as industrial estates.
image_show_water	boolean	Include water items.
image_show_pedestrianarea	boolean	Include pedestrian areas.
image_show_aircraftroad	boolean	Include aircraft roads.
image_show_land	boolean	Include land polygons.

```
<!ELEMENT name_node ( #PCDATA ) >
<!ATTLIST name_node language %language_t; #REQUIRED >
```

Name	Type	Description
language	string	Language of string in the name node.

A name node describes a name in a specific language.

```
<!ELEMENT top_region ( top_region_id, boundingbox, name_node )>
<!ATTLIST top_region top_region_type %top_region_t; #REQUIRED >
<!ELEMENT top_region_id ( #PCDATA )>
```

Name	Type	Description
top_region_id	integer	A unique id for a top region.
boundingbox		The box in which the top region resides.
name_node		The name of the top region

A top region is a high level area such as a country.

7 User Profile

This request and reply allows users to change their profile. New user can also be added. Users cannot be removed through the API due to debiting and security considerations.

These functions are not generally available without certain permissions.

7.1 User Request

```
<!ELEMENT user_request (user)>
<!ATTLIST user_request transaction_id ID #REQUIRED
                       new_user %bool; #IMPLIED >
```

Name	Type	Description
user		See user node doc.
transaction_id	string	Unique transaction id.
new_user	boolean	Whether or not to add a new user.

A user that should be modified in some way or a new users to be added. The *transaction_id* as a unique (within the document) id of the transaction to enable mapping the status in the document sent as reply to the correct transaction.

The new_user attribute tell whether the request is for adding a new user or editing an existing user. If new_user attribute is not set the user will be added if user_id doesn’t already exist and changed if it does. It is recommended to use the new_user attribute.

```
<!ELEMENT user ( user_id, first_name?, last_name?, initials?,
                 language?, measurement_system?,
                 email_address?, operator_comment?,
                 search_for_municipal?, search_for_city?,
                 search_for_citypart?, search_for_zipcode?,
                 search_for_ziparea?,
                 search_for_street?,
                 search_for_company?, search_for_category?,
                 search_for_misc?,
                 new_password?, old_password?,
                 service*, phone*, user_licence_key*,
                 binary_key*, region_access*,
                 wayfinder_subscription?, right*, token*, pin*,
                 id_key*, last_client* )>
<!ATTLIST user uin CDATA #IMPLIED
               birth_date CDATA #IMPLIED
               route_cost %route_cost_t; #IMPLIED
               route_vehicle %route_vehicle_t; #IMPLIED
               search_match_type %matchtype_t; #IMPLIED
               search_word_match_type %wordmatch_t; #IMPLIED
               search_sort_type %sorttype_t; #IMPLIED
               valid_date %time_t; #IMPLIED
               edit_user_right %bool; #IMPLIED
               address1 CDATA #IMPLIED
               address2 CDATA #IMPLIED
               address3 CDATA #IMPLIED
               address4 CDATA #IMPLIED
               address5 CDATA #IMPLIED
               route_turn_image %route_turn_image_t; #IMPLIED
               overview_image_type %overview_image_t; #IMPLIED
               transactionBased %transactionBased_t; #IMPLIED
               deviceChanges %number; #IMPLIED
               supportComment CDATA #IMPLIED
               postalCity CDATA #IMPLIED
               zipCode CDATA #IMPLIED
               companyName CDATA #IMPLIED
               companyReference CDATA #IMPLIED
               companyVATNbr CDATA #IMPLIED
               emailBounces %number; #IMPLIED
               addressBounces %number; #IMPLIED
               customerContactInfo CDATA #IMPLIED >
```

A description of the addition or changes of information for one user. The values for the present elements are updated, the others will be left unchanged. It is not possible to remove a user that has been added, to disable a user access to the $\mathcal{MC}^2$-system all services (see below) should be removed.

Name	Type	Description
uin	integer	The user identification.
birth_date	string	The birth date of the user.
route_cost		The default route cost for user.
route_vehicle		The default route vehicle for user.
search_match_type		The default search string matching method.
search_word_match_type		The default type of word matching.
search_sort_type		The default type of sorting.
valid_date		The time until which the user has access.
edit_user_right	boolean	If the user may edit other users.
address1	string	The 1:st address field.
address2	string	The 2:nd address field.
address3	string	The 3:rd address field.
address4	string	The 4:th address field.
address5	string	The 5:th address field.
route_turn_image		The type of turn images prefered.
overview_image_type		The type of overview image prefered.
transactionBased		The type of transactions used for user. See Section 5.
deviceChanges	integer	The number of device changes the user has, -1 means unlimited number changes.
supportComment	string	The support comment.
postalCity	string	The postal city.
zipCode	string	The zip code.
companyName	string	The company name.
companyReference	string	The company reference.
companyVATNbr	string	The company VAT number.
emailBounces	integer	If the email bounces.
addressBounces	integer	If the address bounces.
customerContactInfo	string	The customer contact information.

<!ELEMENT user_id (#PCDATA)>

A unique identification id of a user.

<!ELEMENT first_name (#PCDATA)>

The first name of a user.

<!ELEMENT last_name (#PCDATA)>

The last name of a user.

<!ELEMENT initials (#PCDATA)>

The initials of a user.

<!ELEMENT language (#PCDATA)>

The language that the user prefers. See section 5 and the language_t entity for valid values.

<!ELEMENT measurement_system (#PCDATA)>

The measurement system for the distances. Valid values are (case insensitive):

Value	Description
metric	The distances will be presented to the user in meters.
imperial	The distances will be presented to the user in foots.

<!ELEMENT email_address (#PCDATA)>

The email address of the user.

<!ELEMENT operator_comments (#PCDATA)>

The operator comments about a user, not to be shown to user.

The *search_for_municipal*, *search_for_city* and *search_for_citypart* represent booleans that sets if search result may include municipals, cities and cityparts.

The *search_for_street*, *search_for_company*, *search_for_category* and *search_for_misc* represent booleans that sets if search result may include streets, companies, categories and miscs.

<!ELEMENT new_password (#PCDATA)>

The new password for user. This will be the future password for user.

<!ELEMENT old_password (#PCDATA)>

The old password for user. This will be checked against the current password user has and if it matches then *new_password* will be set. If *old_password* isn't right then error code -105 is returned.

<!ELEMENT service (service_type, service_method, service_delete?)>

Specifies one service that should be added to or deleted from the user.

<!ELEMENT service_type (#PCDATA)>

The type of service. Valid values are (case sensitive):

ROUTE The service that, given an origin and at least one destination, returns a route description between the start and the destination to the user.

<!ELEMENT service_method (#PCDATA)>

The way used to access the service of a given type. Valid values are (case sensitive):

Value	Description
SMS	The service type specified by the *service_type*-element might be accessed via SMS (Short Message Service).
WAP	The service type specified by the *service_type*-element might be accessed via WAP (Wireless Application Protocol).
HTML	The service may be accessed via html.
NAV	The service may be accessed via proprietary Navigator protocoll.
XML	The service may be accessed via XML(this) api.
OPERATOR	The service may be accessed via operator interface.

<!ELEMENT service_delete EMPTY>

If this element is present the service should be deleted from the user. That is, the user will not have access to the specified service any more.

<!ELEMENT phone (phone_number, phone_manufacturer?, phone_model?,
phone_delete?)>

The information about the mobile phone of the user. If the *phone_number*-element contains the number of an existing phone then this will be changed, otherwise a new phone will be added. ***NB!*** *The support for more than one phone per user might depend on the* auth_user *(could be restricted to one phone per user).*

<!ELEMENT phone_number (#PCDATA)>

The phone number of this phone. Including the country code but without any '+', e.g. "4623456789".

<!ELEMENT phone_manufacturer (#PCDATA)>

The manufacturer of the phone. Valid values are (case insensitive):

Ericsson **Motorola** **Nokia**
Panasonic **Philips** **Siemens**
Default

For other manufacturers you may use the settings for *Default*, that work with most phones. Also select the *Unknown* as model.

<!ELEMENT phone_model (#PCDATA)>

The model of the phone. Valid values depends on the manufacturer and are (case insensitive):

Ericsson Models: 380e, A1018, A2618, GA318, GA628, GF337, GF388, GF788, GH337, GH388, GH688, R250, R310, R320s, R380s, R520s, S868, S888, T18, T20e, T20s, T28, T29, T39, T65, T66 and T68.

Nokia Models: 3310, 3330, 6210, 6250, 6310, 7110, 8210, 8310, 8850, 8890, 9110 and 9210.

Siemens Models C25, C28, C30, C35i, M35i, ME45, S25, S35i, S40, S45 and SL45.

Motorola Models: A008, cd930, M3588, M3888, P7389, T180, T191, T192, T2288, T250, T260, T280, V100, V101, V2288, V3688, V3690 and V66.

Default Models: Unknown.

```
<!ELEMENT phone_delete EMPTY>
```

If this element is present the phone should be deleted from the user.
If it is not present, the phone is added or modified.

```
<!ELEMENT binary_key ( key_data, key_delete? ) >
<!ATTLIST binary_key id CDATA #REQUIRED >
<!ELEMENT key_data ( #PCDATA ) >
<!ELEMENT key_delete EMPTY >
```

This element describes a binary key that is used to verify a user. The key_data is the binary key base64 encoded.
If the key_delete element is present then the binary_key will be removed.
This element is obsolete use user_licence_key instead.

```
<!ELEMENT user_licence_key EMPTY >
<!ATTLIST user_licence_key id       CDATA                 #REQUIRED
                           key      CDATA                 #REQUIRED
                           key_type %hardware_key_type_t; #IMPLIED
                           product  CDATA                 #IMPLIED
                           delete   %bool;                #IMPLIED >
```

This element describes a hardware key that is used to verify a user.
If the key_delete attribute is true is present then the user_licence_key will be removed.

```
<!ELEMENT region_access ( region_access_delete? ) >
<!ATTLIST region_access id           %number; #REQUIRED
                        top_region_id %number; #REQUIRED
                        start_time    %time_t; #REQUIRED
                        end_time      %time_t; #REQUIRED >
<!ELEMENT region_access_delete EMPTY>
```

This element describes an Allowed User Region Access (A.U.R.A.). The region_access says where and when the user may use the server's different regions. If the region_access_delete element is present then the region_access will be removed.

Name	Type	Description
region_access_delete		If present then the specified *region_access* will be removed.
id	integer	Unique id of the region.
top_region_id	integer	The top region for which the access belongs to.
start_time		Start time when the access is valid.
end_time		End time when the access will be invalid.

```
<!ELEMENT wayfinder_subscription ( wayfinder_subscription_delete? )>
<!ATTLIST wayfinder_subscription id      %number; #REQUIRED
                                 type    %number; #REQUIRED >
<!ELEMENT wayfinder_subscription_delete EMPTY>
```

This element describes the level of a wayfinder user's subscription. Obsoleted by *user_rights*. If the wayfinder_subscription_delete element is present then the wayfinder_subscription will be removed.

Name	Type	Description
id	integer	
type	integer	
wayfinder_subscription_delete		If present then the current *wayfinder_subscription* will be removed.

```
<!ELEMENT right EMPTY >
<!ATTLIST right         id            %number; #REQUIRED
                        add_time      %time_t; #IMPLIED
                        type          %number; #IMPLIED
                        top_region_id %number; #IMPLIED
                        start_time    %time_t; #IMPLIED
                        end_time      %time_t; #IMPLIED
                        deleted       %bool;   #IMPLIED
                        origin        CDATA    #IMPLIED >
```

This element describes a right the user has.

Name	Type	Description
id	integer	
add_time		
type	integer	
top_region_id	integer	The right belong to this top region.
start_time		The start time when this right is valid.
end_time		The end time when this right becomes invalid.
delete	boolean	If this rights i deleted or not.
origin	string	A comment string about the origin.

```
<!ELEMENT token EMPTY >
<!ATTLIST token id            %number; #REQUIRED
                create_time   %time_t; #IMPLIED
                age           %number; #IMPLIED
                token         CDATA    #IMPLIED
                group         CDATA    #IMPLIED >
```

This element describes a token the user has received from *activate_request*.

Name	Type	Description
id	integer	
create_time		The creation time for this token.
age		
token	string	The actual token.
group	string	

```
<!ELEMENT pin ( delete? ) >
<!ATTLIST pin id            %number; #REQUIRED
              PIN           CDATA    #IMPLIED
              comment       CDATA    #IMPLIED >
<!ELEMENT delete EMPTY >
```

This element describes a pin the user has to allow other users to access some parts of the user's data. If the delete element is present then the pin will be removed.

Name	Type	Description
id	integer	
PIN	string	The actual PIN.
comment	string	A comment about the PIN.
delete		If present then the pin should be deleted.

```
<!ELEMENT id_key ( delete? ) >
<!ENTITY % id_key_t "(account|hardware|hardware_and_time|
                      service_id_and_time)">
<!ATTLIST id_key id           %number;      #REQUIRED
                 type         %id_key_t;    #IMPLIED
                 key          CDATA         #IMPLIED >
```

This element describes an key which identifies a user. If the delete element is present then the id_key will be removed.

Name	Type	Description
id	integer	
type		Key type.
key	string	The key.
delete		If present then the key should be removed

```
<!ELEMENT last_client EMPTY >
<!ATTLIST last_client id                  %number; #REQUIRED
                      client_type         CDATA    #IMPLIED
                      client_type_options CDATA    #IMPLIED
                      version             CDATA    #IMPLIED
                      extra               CDATA    #IMPLIED
                      origin              CDATA    #IMPLIED
                      history             %bool;   #IMPLIED
                      changer_uin         CDATA    #IMPLIED
                      change_time         %time_t; #IMPLIED >
```

This element describes what client types a user has and has had.

Name	Type	Description
id	integer	
client_type	string	Last client type used.
client_type_options	string	
version	string	
extra	string	
origin	string	
history	boolean	
changer_uin	integer	
change_time		When the last change was made.

```
<!ELEMENT favorite ( position_item, fav_info* )>
<!ATTLIST favorite
                    id                CDATA  #REQUIRED
                    name              CDATA  #REQUIRED
                    short_name        CDATA  #REQUIRED
                    description       CDATA  #REQUIRED
                    category          CDATA  #REQUIRED
                    map_icon_name     CDATA  #REQUIRED >
<!ELEMENT fav_info EMPTY >
<!ATTLIST fav_info  type  %poi_info_t; #REQUIRED
                    key   CDATA        #REQUIRED
                    value CDATA        #REQUIRED >
```

Favorite describes a special place that the user commonly wants to go to.

Name	Type	Description
favorite		A favorite place for the user.
id		The id that is used to identify the favorite. See also Section 8.1 *favorite_id*.
name	string	The name of the favorite to show in selection list.
short_name	string	A short name of the favorite for quick selection. Might not be available on all interfaces.
description	string	A text describing the favorite and/or additional information.
category	string	Used to group favorites together. Currently not used.
map_icon_name	string	The symbol to use for the favorite when drawn on maps. Currently not used.
fav_info		The list of information elements for the favorite.

7.2 User Reply

The following error codes may also be returned apart from the generic ones:

Code	Description
-105	Old password not valid. The supplied *old_password* is wrong.
-106	Must supply old password. *old_password* is required when changing password.
-107	Not unique userID. A new user must have a unique user_id.
-108	No such user. Could not find user to edit, check user_id.

```
<!ELEMENT user_reply (status_code, status_message, status_code_extended?)>
<!ATTLIST user_reply transaction_id ID #REQUIRED>
```

The status code and message with the result of the request. The *transaction_id* is the ID of the transaction that this reply refers to.

Name	Type	Description
status_code	integer	See table.
status_message	string	A message describing the error
status_code_extended		Extended code.

8 User Favorites

Request for add, delete and synchronize user's favorites. It is also possible to set a "auto route" favorite. These functions are not generally available without certain permissions.

8.1 User Favorites Request

```
<!ELEMENT user_favorites_request ( (user_id | uin |
                                    (user_session_id, user_session_key) )?,
                                   favorite_id_list?,
                                   delete_favorite_id_list?,
                                   add_favorite_list?,
                                   auto_dest_favorite? )>
<!ATTLIST user_favorites_request
                               transaction_id ID #REQUIRED
                               fetch_auto_dest   %bool; "false"
                               sync_favorites    %bool; "true"
                               position_system   %position_system_t; "MC2"
                               fav_info_in_desc  %bool; "true" >
<!ELEMENT favorite_id_list ( favorite_id* )>
<!ELEMENT favorite_id ( #PCDATA )>
<!ELEMENT delete_favorite_id_list ( favorite_id+ )>
<!ELEMENT add_favorite_list ( favorite+ )>
<!ELEMENT auto_dest_favorite ( favorite? )>
```

Name	Type	Description
user_id uin user_session_id user_session_key		The user of the favorites. Identified by *user_id* or *user_session* or *uin*.
favorite_id_list		The id's of the favorites that the client has.
delete_favorite_id_list		The id's of the favorites that the client has deleted.
add_favorite_list		The favorites that the client has added.
auto_dest_favorite		The "auto route" favorite. If the *auto_dest_favorite* element is present and empty then the "auto route" favorite is cleared.
fetch_auto_dest		If the "auto route" favorite should be sent in the reply.
sync_favorites		If the reply should contain the add and delete favorites that is needed to synchronize the client, uses *favorite_id_list*. Otherwise the reply only contains the added and deleted favorites in the request.
position_system		The position_system to use in the favorites in the reply.
fav_info_in_desc		If *fav_info*s should be added to *description* attribute. New clients that support *fav_info*s should set this to *false*.
favorite_id		An id of a favorite. Value is from an *id* in a *favorite* described in Section 7.1.

8.2 User Favorites Reply

```
<!ELEMENT user_favorites_reply ( (delete_favorite_id_list?,
                                  add_favorite_list?,
                                  auto_dest_favorite?) |
                                 ( status_code, status_message,
                                   status_code_extended? ) )>
<!ATTLIST user_favorites_reply transaction_id ID #REQUIRED
                               crc CDATA #REQUIRED >
```

Name	Type	Description
delete_favorite_id_list		The id's of the favorites that the client should remove from it's list.
add_favorite_list		The favorites that the client should add to it's list.
auto_dest_favorite		The "auto route" favorite or empty if no "'auto route" favorite.
crc		the favorites crc. Can be used with *user_favorites_crc_request*.

8.3 User Favorites CRC

Calculates crc for the user favorite list.

8.3.1 User Favorites CRC Request

```
<!ELEMENT user_favorites_crc_request EMPTY >
<!ATTLIST user_favorites_crc_request transaction_id ID #REQUIRED
                                     crc CDATA #REQUIRED >
```

Request for crc match.

Name	Type	Description
crc		Match against favorites crc.

8.3.2 User Favorites CRC Reply

```
<!ELEMENT user_favorites_crc_reply EMPTY >
<!ATTLIST user_favorites_crc_reply transaction_id ID #REQUIRED
                                   crc_match %bool; #REQUIRED >
```

Name	Type	Description
crc_match	boolean	True if the crc matched.

9 User Capabilities

Request for user capabilities.

9.1 User Cap Request

```
<!ELEMENT user_cap_request EMPTY >
<!ATTLIST user_cap_request transaction_id ID #REQUIRED >
```

A request for user capabilities.

9.2 User Cap Reply

```
<!ELEMENT user_cap_reply (user_id, cap*, pin*, popup*) >
<!ATTLIST user_cap_reply transaction_id ID #REQUIRED >
```

```
<!ENTITY % cap_name_type "(gps|locator|route|fleet|traffic)">
<!ELEMENT cap EMPTY >
<!ATTLIST cap name %cap_name_type; #REQUIRED >
<!ELEMENT popup (popup_message, popup_once?, popup_url?)>
<!-- Yes No if url and attr for if to exit if no on url. -->
<!ENTITY % popup_url_t "(yes_no|goto_or_exit)">
<!ELEMENT popup_message ( #PCDATA )>
<!ELEMENT popup_once ( #PCDATA )>
<!ELEMENT popup_url ( #PCDATA )>
<!ATTLIST popup_url url_type %popup_url_t; "yes_no" >
```

The reply to a *user_cap_reply* with the *user_id*, *cap* and *pin*. The popup element indicates that the client must show the message with an ok button if the isn't an url. If the is an url then the client shows yes and no buttons. If yes url is opened. If no check the *url_type* attribute if exit application ("goto_or_exit") or continue to main menu ("yes_no").

Name	Type	Description
popup		
popup_message	string	Message to be shown in the popup.
popup_once		If node is present then the popup should only be shown once.
popup_url	string	URL to be shown if the clients accepts it (by pressing "yes")
url_type		What kind of user choice buttons to show.

```
<!ELEMENT cap EMPTY >
<!ATTLIST cap name %cap_name_type; #REQUIRED >
```

The users capability, for example "*gps*" or "*route*".

10 User Show

Request for the data about a user.

10.1 User Show Request

```
<!ELEMENT user_show_request ( (user_id |
                               (user_session_id, user_session_key))? ) >
<!ATTLIST user_show_request transaction_id ID #REQUIRED
                            uin CDATA #IMPLIED >
```

A request for a specific user's settings and data. The user can be identified by *user_id* or by *user_session_id* and *user_session_key* or by an *uin*.

Name	Type	Description
uin	integer	if present, used to find user even if *user_id* or *user_session_id* and *user_session_key* is present.

10.2 User Show Reply

```
<!ELEMENT user_show_reply ( ( user ) |
                            ( status_code, status_message,
                              status_code_extended? ) )>
<!ATTLIST user_show_reply transaction_id ID #REQUIRED>
```

The reply to a *user_show_request* with the user data or a status code and message expaining why not.

Name	Type	Description
user		Contains user data information nodes.
status_code	integer	Error code.
status_message	string	Error message.
status_code_extended	integer	Extended status code. For additional errors.

11 Route

11.1 Route Request

```
<!ELEMENT route_request ( route_request_header,
                          routeable_item_list,
                          routeable_item_list )>
```

A route request with origin(s) and destination(s) to make a route between.

Name	Type	Description
route_request_header		Settings for route.
routeable_item_list		First node is origins for route and the second node is destinations.

```
<!ATTLIST route_request transaction_id ID #REQUIRED>
```

The unique identifier of the *route_request*.

```
<!ELEMENT route_request_header ( route_preferences )>
<!ENTITY % reroute_reason_t "(unknown|truncated_route|off_track|
                             traffic_info_update|user_request)" >
<!ATTLIST route_request_header
          previous_route_id CDATA               #IMPLIED
          reroute_reason    %route_reason_t;    #IMPLIED>
```

The settings of the *route_request* are in this element.

Name	Type	Description
previous_route_id	string	Shall be included when the route_request is actually a reroute of a previous route.
reroute_reason		Describes why a reroute was necessary. Using a correct *reroute_reason* value will give better service.

Values for reroute_reason:

Value	Description
unknown	When the reason is not known.
truncated_route	If the client downloaded a truncated route this reason shall be used when requesting the next part of the route. The XML server never delivers truncated routes.
off_track	The user has left the route and requires a new one.
traffic_info_update	The client want's the route updated with current traffic information.
user_request	The user requested a reroute.

```
<!ELEMENT route_preferences ( ( user_id | route_settings | uin |
                                (user_session_id, user_session_key) ),
                              image_settings? )>
```

The route settings of a user can be used or the settings can be explicitly entered. The user can be identified by *user_id* or *user_session_id* and *user_session_key* or *uin*. Optionally settings for images can be set with an *image_settings* element.

```
<!ENTITY % route_description_type_t "(normal|compact)">
```

The type of route descriptions to get. Valid values are (case sensitive):

Value	Description
normal	The normal type of description with a natural flow of text.
compact	The compact type of description which is for devices with a limited display, such as some cellulars. The text in this case will be abbreviated in order for it to fit into a narrow cellular display. Also, some messaging protocols limit the amount of text that may be transferred per message. SMSes for example, have a limit of 160 characters.

The following short except from a route description will illustrate the difference made by different values of *route_description_type_t*. *route_description_type* set to *normal* gives

```
Drive 46 meters then turn right into the 1:st street Karsviksgatan
Drive 180 meters then turn left into the 1:st street Pontonjärgatan
Drive 130 meters then turn right into the 2:nd street Sven Rinmans gata
```

route_description_type set to *compact* gives

```
46m 1->
Karlsviksgatan
180m 1<-
Pontonjärgatan
130m 2->
Sven Rinmans Gata
```

Abbreviations here are to be interpreted as first the number of streets to pass, and then the turn direction, so that “2->” is interpreted as “turn right into the 2:nd street”. Street names may also be abbreviated using the *abbreviate_route_names* flag.

```
<!ATTLIST route_preferences
          route_description_type %route_description_type_t; #REQUIRED
          route_image_links %bool; "false"
          route_overview_image_width %number; "256"
          route_overview_image_height %number; "256"
          route_turn_image_width %number; "256"
          route_turn_image_height %number; "256"
          route_image_default_format %route_image_format_t; "png"
          route_image_display_type %image_display_type; "std"
          route_turn_data %bool; "false"
          route_boundingbox_position_sytem %position_system_t; "MC2"
          route_turn_boundingbox %bool; "false"
          route_road_data %bool; "false"
          route_items %bool; "true"
          abbreviate_route_names %bool; "true"
          route_landmarks %bool; "false" >
```

The attributes for the route preferences.

Name	Type	Description
route_description_type		The preference for the type of route description is in this attribute
route_image_links	boolean	Set if route-turn-image links should be added to the reply.
route_overview_image_width	integer	The width of the route overview image.
route_overview_image_height	integer	The height of the route overview image.
route_turn_image_width	integer	The width of the turn images.
route_turn_image_height	integer	The height of the turn images.
route_image_default_format		Image format, used if the HTTP request doesn't contain any supported image format in it's Accept header line.
route_image_display_type		The type of display that should display the images.
route_turn_data	boolean	Set if the building blocks of the description should be sent. This adds the turn, distance, time, road name, exit count, signpost text, signpostexitnbr, signpostroutenbr, transportation_type, crossing_type element. For the first turn, the start_dir and route_housenumber_start_direction elements are added. Use this flag if you want the elements used for each turn instruction.
route_boundingbox_position_system		The positioning system for the turn boundingbox and road data.
route_turn_boundingbox	boolean	Set if a boundingbox for the turns should be added.
route_road_data	boolean	Set if route road data should be added to the reply. This includes coordinates, and may become a huge amount of data. Make sure you really need this before using it.
route_items	boolean	Set if route_items should be added to reply.
abbreviate_route_names	boolean	Set if the street names should be abbreviated or not. Default is true.
route_landmarks	boolean	If landmarks should be included in the reply. Default is false.

```
<!ELEMENT route_settings ( route_costA?,
                           route_costB?,
                           route_costC?,
                           language )>
```

The settings to use when routing, see below for an explanation of the parameters.

```
<!ATTLIST route_settings route_vehicle %route_vehicle_t; #REQUIRED
                         avoid_toll_road %bool; #IMPLIED
                         avoid_highway   %bool; #IMPLIED >
```

Type of transportation to use for the route. The *avoid_toll_road* attribute is if toll-roads should be penalised when calculating the route. The *avoid_highway* attribute is if highways should be penalised when calculating the route.

```
<!ELEMENT route_costA ( #PCDATA )>
```

If the value isn’t “0” then the distance to go from origin to destination is a parameter to minimize when calculating the route.

```
<!ELEMENT route_costB ( #PCDATA )>
```

If the value isn’t “0” then the time between origin and destination is a parameter to minimize when calculating the route.

```
<!ELEMENT route_costC ( #PCDATA )>
```

If the value isn’t “0” then the time to go from origin to destination taking traffic, road and weather disturbances into account is a parameter to minimize when calculating the route.

```
<!ELEMENT routeable_item_list ( (position_item | search_item)+ )>
```

A list of items to use as origin or destination for the route.

11.2 Route Reply

The status codes that may be returned in a route reply. In addition to the generic ones like “-5” “Outside allowed area”.

Code	Description
-501	No route found. No route was found from the origin to the destination. Example: origin is on an island with no bridges and no information on ferries.
-502	Too far for vehicle. Route is too far to go for the vehicle used. Mostly used for pedestrian routes that are too long.
-503	Problem with origin. Can not make out origin. Example: origin position is too far from a drivable/walkable street. Or an invalid search_item was sent in the routeable_item_list.
-504	Problem with destination. Can not make out destination. For examples see -503.
-505	Keep your route, it is up to date. This happens when reroute_reason is traffic_info_update and the route is unchanged.
-94209	Routing not allowed. You need to buy route service. This error can be sent to Content Window. Number is 0x17001.

```
<!ELEMENT route_reply ( ( route_reply_header,
                          route_origin, route_destination,
                          route_reply_items ) |
                        ( status_code, status_message, status_uri?,
                          status_code_extended? ) )>
```

The result of a *route_request* with the route description, or a status code and a message with an error description.

Name	Type	Description
route_reply_head		Statistics of the route.
route_origin		Contains the route origin(s).
route_destination		Contains the route destination(s).
route_reply_items		The route as number of elements. If the *route_items* attribute in the *route_preferences* is set to false then this element is empty.

```
<!ATTLIST route_reply transaction_id ID       #REQUIRED
                      route_id       CDATA    #REQUIRED
                      ptui           %number; #IMPLIED >
```

Name	Type	Description
route_id	string	Unique id for the route in the $\mathcal{MC}^2$ system that can be used later as a reference to the route.
ptui	integer	Periodic traffic info interval in minutes.

```
<!ELEMENT route_reply_header ( total_distance,
                               total_distance_nbr,
                               total_time,
                               total_time_nbr,
                               total_standstilltime,
                               total_standstilltime_nbr,
                               average_speed,
                               average_speed_nbr,
                               routing_vehicle,
                               routing_vehicle_type,
                               boundingbox,
                               route_overview_link?,
                               route_overview_width?,
                               route_overview_height? )>
```

The header of the route reply with statistics of the route.

```
<!ELEMENT total_distance ( #PCDATA )>
```

The total distance of the route, in the form of a string with a measurement unit and a short explanation, for example “*Total distance: 5.2 km*”.

```
<!ELEMENT total_distance_nbr ( #PCDATA )>
```

The total distance of the route in meters, for example “*5198*”.

```
<!ELEMENT total_time ( #PCDATA )>
```

The total time of the route, formatted as a string, such as “*Total time: hh:mm:ss*”.

```
<!ELEMENT total_time_nbr ( #PCDATA )>
```

The total time of the route in seconds, for example “*253*”.

```
<!ELEMENT total_standstilltime ( #PCDATA )>
```

The total standstill time of the route, formatted as a string, such as “*Standstill time: hh:mm:ss*”.

```
<!ELEMENT total_standstilltime_nbr ( #PCDATA )>
```

The total standstill time of the route in seconds, for example “*25*”.

<!ELEMENT average_speed (#PCDATA)>

The average speed of the route, formatted as a string, such as “*Average speed x km/h*”.

<!ELEMENT average_speed_nbr (#PCDATA)>

The average speed of the route in meters per second, for example “*11.525097*”.

<!ELEMENT routing_vehicle (#PCDATA)>

The routing vehicle of the route, formatted as a string, using the language, of the request, such as “*passenger car*” or “*pedestrian*”.

<!ELEMENT routing_vehicle_type (#PCDATA)>

The routing vehicle of the route as a *route_vehicle_t*, for example “*passengercar*” or “*pedestrian*”.

<!ELEMENT route_overview_link (#PCDATA)>

A URI to the overview image of the route.

<!ELEMENT route_overview_width (#PCDATA)>

The width of the overview image, in pixels, formatted as a string. This is normally the same as the requested image width, but may also be less.

<!ELEMENT route_overview_height (#PCDATA)>

The height of the overview image, in pixels, formatted as a string. This is normally the same as the requested image height, but may also be less.

<!ELEMENT route_origin (search_item+)>

The origin(s) of the route.

```
<!ELEMENT route_destination ( search_item+ )>
```

The destination(s) of the route.

```
<!ELEMENT route_reply_items ( route_reply_item* )>
```

The route as a number of elements. If the *route_items* attribute in the *route_preferences* is set to false then this element is empty.

```
<!ELEMENT route_reply_item ( description?,
                             turn?,
                             distance?,
                             time?,
                             roadname?,
                             exitcount?,
                             signposttext?,
                             signpostexitnbr?,
                             signpostroutenbr?,
                             start_dir?,
                             route_housenumber_start_direction?,
                             transporation_type?,
                             crossing_type?,
                             route_turn_link?,
                             route_turn_width?,
                             route_turn_height?,
                             boundingbox?,
                             position_item?,
                             route_road_item*,
                             route_landmark_item* )>
<!ATTLIST route_reply_item   controlled_access   %bool; #IMPLIED
                             ramp               %bool; #IMPLIED
                             roundabout         %bool; #IMPLIED
                             drive_on_right_side %bool; #IMPLIED >
```

A part of the route describing a turn or some other action. The controlled_access, ramp, roundabout and drive_on_right_side attributes here are for the road near the turn not necessarily valid for the entire distance to the next turn. If the attributes are not present then they have the same value as the last time they appeared in a *route_reply_item* or *route_road_item*. This means that for example if drive_on_right_side is set in the first *route_reply_item* to true and not in any following *route_reply_item* or *route_road_item* the entire journey from start to end there is right-hand traffic.

```
<!ELEMENT description ( #PCDATA )>
```

The description of the *route_reply_item*, as a human readable driving direction text string.

```
<!ENTITY % route_turn_t "(left|right|ahead|u_turn|followroad|
                         enter_roundabout|exit_roundabout|
                         ahead_roundabout|right_roundabout|
                         left_roundabout|off_ramp|on_ramp|
                         enter_bus|exit_bus|change_bus|
                         park_car|start|finally|exit|
                         keep_left|keep_right|
                         enter_ferry|exit_ferry|change_ferry|
                         start_with_u_turn|u_turn_roundabout|
                         endofroad_left_turn|endofroad_right_turn|
                         off_ramp_left|off_ramp_right|
                         on_main|off_main|
                         no_turn)">
<!ENTITY % crossing_t "undefined_crossing|no_crossing|
                       crossing_3ways_t|crossing_3ways_y|crossing_4ways|
                       crossing_5ways|crossing_6ways|crossing_7ways|
                       crossing_8ways|crossing_2roundabout|
                       crossing_3roundabout|crossing_4roundabout|
                       crossing_4roundabout_asymmetric|
                       crossing_5roundabout|crossing_6roundabout|
                       crossing_7roundabout" >
<!ENTITY % route_start_dir_t "(north|northnortheast|northeast|
                               eastnortheast|east|eastsoutheast|
                               southeast|southsoutheast|south|
                               southsouthwest|southwestwestsouthwest|
                               west|westnorthwest|northwest|
                               northnorthwest)" >
<!ENTITY % route_housenumber_start_direction_t "(leftodd|rightodd|
                                                 increasing|decreasing|
                                                 unknown)">
<!ENTITY % route_transportation_t "(drive|walk|bus)">
```

The turn, crossing, housenumber starting direction and transportation method types.

route_turn_t type of turn or other action:

Value	Description
left	A normal left turn.
right	A normal right turn.
ahead	Drive straight ahead, for example in a crossing.
u_turn	Make a U-turn, that is, turn the vehicle around.
followroad	Follow the current road.
enter_roundabout	Enter into a roundabout.
exit_roundabout	Exit out of a roundabout.
ahead_roundabout	Drive straight ahead at the roundabout.
right_roundabout	Make a right turn at the roundabout.
left_roundabout	Make a left turn at the roundabout.
off_ramp	Take an off ramp, for example to get off a highway.
on_ramp	Take an on ramp, for example to get on a highway.
enter_bus	Enter a bus. For pedestrians.
exit_bus	Exit a bus. For pedestrians.
change_bus	Get off the current bus, and get on another one.
park_car	Park the car and continue as a pedestrian.
start	Route starts here.
finally	Destination is straight ahead in the current direction.
exit	Drive of current road.
keep_left	Make a slight left turn.
keep_right	Make a slight right turn.
enter_ferry	Enter a ferry.
exit_ferry	Exit a ferry.
change_ferry	Change to another ferry.
start_with_u_turn	Route starts here but you have to turn the vehicle around first.
u_turn_roundabout	Drive around in the roundabout exiting in the same direction as entering.
endofroad_left_turn	Take left road where road ends.
endofroad_right_turn	Take right road where road ends.
off_ramp_left	Take an off ramp to the left.
off_ramp_right	Take an off ramp to the right.
off_main	Take an off ramp, off non highway road.
on_main	Take an on ramp, on non highway road.
no_turn	Not really a turn.

crossing_t:

Value	Description
undefined_crossing	The crossing is handled in an alternative way.
no_crossing	There is no crossing here “Follow the road”.
crossing_3ways_t	Crossing of three roads in the shape of a “T”.
crossing_3ways_y	Crossing of three roads in the shape of a “Y”.
crossing_4ways	Crossing of four roads.
crossing_5ways	Crossing of five roads.
crossing_6ways	Crossing of six roads.
crossing_7ways	Crossing of seven roads.
crossing_8ways	Crossing of eight roads.
crossing_2roundabout	Crossing is a roundabout with only two exits.
crossing_3roundabout	Crossing is a roundabout with three exits.
crossing_4roundabout	Crossing is a roundabout with four exits.
crossing_4roundabout_asymmetric	Crossing is a roundabout with four exits. The four exits however, are not positioned in a symmetric way.
crossing_5roundabout	Crossing is a roundabout with five exits.
crossing_6roundabout	Crossing is a roundabout with six exits.
crossing_7roundabout	Crossing is a roundabout with seven exits.

route_start_dir_t route_housenumber_start_direction_t:

Value	Description
leftodd	House numbers on left side of the vehicle should be odd.
rightodd	House numbers on right side of the vehicle should be odd.
increasing	House numbers should be increasing in the direction of travel.
decreasing	House numbers should be decreasing in the direction of travel.
unknown	

route_transportation_t:

Value	Description
drive	Current mode of transportation is by car.
walk	Current mode of transportation is by foot.
bus	Current mode of transportation is by bus.

<!ELEMENT turn (#PCDATA)>

One of the *route_turn_t* turn types.

<!ELEMENT distance (#PCDATA)>

The distance for the description in meters. This field is only digits, such as “2166”. The *distance* element is not present for the starting *route_reply_item*. Instead, the starting *route_reply_item* will have a *start_dir* tag.

<!ELEMENT time (#PCDATA)>

The estimated time the distance should take, in seconds, such as “138”. The *distance* element is not present for the starting *route_reply_item*. Instead, the starting *route_reply_item* will have a *route_housenumber_start_direction* tag.

<!ELEMENT roadname (#PCDATA)>

The name of the street that the turn is into, such as “Oxford Street”.

<!ELEMENT exitcount (#PCDATA)>

Indicates how many exits there are on the same side as the road to turn into before the road to turn into. Simply a number, such as “0” or “8”.

<!ELEMENT signposttext (#PCDATA)>

If there is a signpost at the turn this string contains the text on it. An example is “STOCKHOLM, (31), (33)”, which means that the road leads to Stockholm and road numbers are 31 and 33.

<!ELEMENT signpostexitnbr (#PCDATA)>

If there is a signpost at the turn this string contains the exit number on it. A number, only, such as “26”.

<!ELEMENT signpostroutenbr (#PCDATA)>

If there is a signpost at the turn, this string contains the road numbers that the signpost says the turn will lead you to, such as “33”.

<!ELEMENT start_dir (#PCDATA)>

Only present for the starting *route_reply_item*, the starting direction as one of the *route_start_dir_t* types.

<!ELEMENT route_housenumber_start_direction (#PCDATA)>

The house numbering at the start of the route, if available, is one of the *route_housenumber_start_direction_t* types.

<!ELEMENT transporation_type (#PCDATA)>

The transportation method for the *route_reply_item*, as one of the *route_transportation_* types.

<!ELEMENT crossing_type (#PCDATA)>

The type of crossing at the turn, as one of the *crossing_t* types.

<!ELEMENT route_turn_link (#PCDATA)>

The URI for the image covering the turn.

<!ELEMENT route_turn_width (#PCDATA)>

The width of the turn image, in pixels.

<!ELEMENT route_turn_height (#PCDATA)>

The height of the turn image, in pixels.

```
<!ELEMENT boundingbox EMPTY>
```

The boundingbox covering the turn. The data is in the *ATTLIST* of the *boundingbox* element, see section 6.

The *position_item* is sent along with the boundingbox and contains the exact position of the turn's exit.

```
<!ELEMENT route_road_item ( (lat, lon)+ ) >
<!ATTLIST route_road_item
                            speedLimit %number; #REQUIRED
                            is_turn %bool;      #IMPLIED
                            controlled_access   %bool; #IMPLIED
                            ramp                %bool; #IMPLIED
                            roundabout          %bool; #IMPLIED
                            drive_on_right_side %bool; #IMPLIED >
```

The coordinates describing the shape of this part of the route, and other detailed information about the current section of the route. If the attributes are not present then they have the same value as the last time they appeared in a route_reply_item or route_road_item.

Value	Description
speedLimit	The speed limit of the road, 0 is unknown speed.
is_turn	If the road is where the actual turning is done.
controlled_access	If the road has controlled access.
ramp	If the road is a ramp.
roundabout	If the road is part of a roundabout.
drive_on_right_side	If to drive on the right side of the road.

```
<!ELEMENT route_landmark_item ( description, road_side?,
                                landmarklocation_type?, landmark_type?,
                                distance?, name? )>
<!ATTLIST route_landmark_item at_turn   %bool; #REQUIRED
                              is_detour %bool; #IMPLIED
                              is_start  %bool; #IMPLIED
                              is_stop   %bool; #IMPLIED>
<!ELEMENT road_side ( #PCDATA )>
<!ELEMENT landmarklocation_type ( #PCDATA )>
<!ELEMENT landmark_type ( #PCDATA )>
```

Describes a landmark that is passed before the turn or near the turn itself. The *road_side*, *landmarklocation_type*, *landmark_type*, *distance* and *name* is in the reply if *route_turn_data* is "true".

Value	Description
at_turn	If the landmark is adjacent to the turn.
is_detour	If the landmark is a traffic caused detour.
is_start	If the landmark is the begining of a traffic situation.
is_stop	If the landmark is the end of a traffic situation.

12 Search

12.1 Search Request

```
<!ELEMENT search_request ( search_request_header,
                           (search_query | proximity_query ) )>
```

A text search for search areas and search items or a proximity request.

Name	Type	Description
search_request_header		Search settings.
search_query		item and area query.
proximity_query		Position search.

```
<!ATTLIST search_request transaction_id ID #REQUIRED>
```

The unique identifier of the *search_request*.

```
<!ELEMENT search_request_header (search_preferences,
                                 search_explicit_itemid?)>
<!ATTLIST search_request_header
                          position_sytem %position_system_t; "MC2"
                          position_search_items %bool; "false"
                          position_search_areas %bool; "false"
                          search_area_starting_index %number; "0"
                          search_area_ending_index %number; "49"
                          search_item_starting_index %number; "0"
                          search_item_ending_index %number; "99"
                          full_search_area_match_purge %bool; #IMPLIED >
<!ELEMENT search_explicit_itemid EMPTY>
```

Header of a *search_request* which contains search settings.

Name	Type	Description
search_explicit_itemid	string	Obsolete and should not be used, use *position_search_items* instead. If set, then the *explicit_itemid* is added to the returned *search_item*s.
position_search_items	boolean	If “true”, the *position_sytem* is used as the coordinate system in the resulting *search_item*s.
position_search_areas	boolean	If “true”, the *position_sytem* is used as the coordinate system in the resulting *search_area*s.
search_area_starting_index	integer	Start offset for *search_area*s in the search.
search_area_ending_index	integer	End offset for *search_area*s in search.
search_item_starting_index	integer	Start offset for *search_item*s in the search.
search_item_ending_index	integer	End offset for *search_item*s in the search.
full_search_area_match_purge	boolean	Enables full *search_area* match purge. If the *search_area_query* matches a *search_area* fully, “Lund” matches fully “Lund” but not “Lunde”, all other *search_area* matches are removed and a *search_item_query* is performed if possible. This feature is default false.

```
<!ELEMENT search_preferences ( ((user_id|uin), search_settings?) |
                               search_settings |
                               (user_session_id, user_session_key,
                                search_settings?) )>
```

The search settings of a user can be used by specifying a *user_id* element or with a *user_session_id* and *user_session_key* element, or by *uin* element, or they can be explicitly entered with a *search_settings* element. When specifying search settings with an user. Search settings may be overridden or specified by a *search_settings* element after the user.

```
<!ELEMENT search_settings ( search_for_municipal?, search_for_city?,
                            search_for_citypart?, search_for_zipcode?,
                            search_for_ziparea?,
                            search_for_street?,
                            search_for_company?, search_for_category?,
                            search_for_misc?,
                            show_search_area_municipal?,
```

```
                        show_search_area_city?,
                        show_search_area_city_part?,
                        show_search_item_municipal?,
                        show_search_item_city?,
                        show_search_item_city_part?,
                        show_search_item_zipcode?,
                        show_search_item_ziparea?,
                        language? )>
```

The explicit search settings element. The language is the preferred language of the name in *search_item*s or *search_area*s. This setting is used for those items with different names in different languages. For example, the English: Gothenburg and the Swedish: Göteborg refers to the same city.

Name	Type	Description
search_for_municipal		include municipals in search.
search_for_city		Include cities in search.
search_for_citypart		Include city-parts in search.
search_for_zipcode		Include zipcodes in search.
search_for_ziparea		Include zipareas in search.
search_for_street		Include streets in search.
search_for_company		Include companies and points of interest in search.
search_for_category		Include categories of companies and other items in search.
search_for_misc		Include misc items in search.
show_search_area_municipal		Show municipal for search areas.
show_search_area_city		Show city for search areas.
show_search_area_city_part		Show city part for search areas.
show_search_item_municipal		Show municipal for search items.
show_search_item_city		Show city for search items.
show_search_item_city_part		Show city part for search items.
show_search_item_zipcode		Show zipcode for search items.
show_search_item_ziparea		Show ziparea for search items.

```
<!ATTLIST search_settings
                        matchtype %matchtype_t; #IMPLIED
                        wordmatch %wordmatch_t; #IMPLIED
                        sorttype  %sorttype_t;  #IMPLIED
                        minimum_numberhits %number; #IMPLIED >
```

The attributes to *search_settings* is the type of string matching method, the type of word matching and the type of sorting.

The *minimum_numberhits* attribute is how many matches there should at least be in the reply. If the search results in less than *minimum_numberhits* matches the search area is expanded and the search criteria reduced gradually until the number of matches is at least *minimum_numberhits*. Even after expanding the search area and reducing the search criteria

maximally the resulting number of matches may still be less than *minimum_numberhits* in the reply.

<!ELEMENT search_for_municipal EMPTY>

If present in the *search_settings* the search result may include municipals.

<!ELEMENT search_for_city EMPTY>

If present in the *search_settings* the search result may include cities.

<!ELEMENT search_for_citypart EMPTY>

If present in the *search_settings* the search result may include city-parts.

<!ELEMENT search_for_zipcode EMPTY>

If present in the *search_settings* the search result may include zipcodes.

<!ELEMENT search_for_ziparea EMPTY>

If present in the *search_settings* the search result may include zipareas.

<!ELEMENT search_for_street EMPTY>

If present in the *search_settings* the search result may include streets.

<!ELEMENT search_for_company EMPTY>

If present in the *search_settings* the search result may include companies and points of interest.

<!ELEMENT search_for_category EMPTY>

If present in the *search_settings* the search result may include categories of companies and other items.

<!ELEMENT show_search_area_municipal EMPTY>

If present in the *search_settings* the search result *search_area*s should have the municipal they’re in.

<!ELEMENT show_search_area_city EMPTY>

If present in the *search_settings* the search result *search_area*s should have the city they’re in.

<!ELEMENT show_search_area_city_part EMPTY>

If present in the *search_settings* the search result *search_area*s should have the city part they’re in.

<!ELEMENT show_search_area_zipcode EMPTY>

If present in the *search_settings* the search result *search_area*s should have the zipcode they’re in.

<!ELEMENT show_search_area_ziparea EMPTY>

If present in the *search_settings* the search result *search_area*s should have the ziparea they’re in.

<!ELEMENT show_search_item_municipal EMPTY>

If present in the *search_settings* the search result *search_item*s should have the municipal they’re in.

<!ELEMENT show_search_item_city EMPTY>

If present in the *search_settings* the search result *search_item*s should have the city they’re in.

<!ELEMENT show_search_item_city_part EMPTY>

If present in the *search_settings* the search result *search_item*s should have the city part they’re in.

<!ELEMENT show_search_item_zipcode EMPTY>

If present in the *search_settings* the search result *search_item*s should have the zipcode they're in.

```
<!ELEMENT show_search_item_ziparea EMPTY>
```

If present in the *search_settings* the search result *search_item*s should have the ziparea they're in.

```
<!ELEMENT search_query ( top_region?, (search_area_query | search_area),
                         search_item_query? )>
```

If a *top_region* is present then the search is done in that top region if not then the default top region is used to search in. If *search_area_query* is given, the *search_query* is interpreted as a search for an area and optionally an item in that area. If you already have a *search_area*, you may search for an item using the *search_item_query* within that area.

```
<!ELEMENT search_area_query ( #PCDATA )>
```

The name of the area(s) to search for.

```
<!ELEMENT search_item_query ( #PCDATA )>
<!ATTLIST search_item_query house_number CDATA #IMPLIED >
```

The name of the item(s) to search for. Optionally the house number in separate *house_number* attribute.

```
<!ELEMENT proximity_query ( ( ((search_item | position_item), distance?) |
                              boundingbox),
                            search_item_query? ) >
```

A proximity query is used to get *search_item*s in an area. The *search_item* or the *position_item* defines the center point of a circle with *distance* as radius, in metres. This circle defines the area to search in. If *distance* is left out the whole country is searched and the closest matches are returned. The *search_item_query*, if present, is used as in a normal *search_query* to find only those results that matches the string.

12.2 Search Reply

```
<!ELEMENT search_reply ( ( search_area_list?, search_item_list? ) |
                         ( status_code, status_message,
                           status_code_extended? ) )>
```

The result of a *search_request* with lists of matches to the search request.

```
<!ATTLIST search_reply transaction_id ID #REQUIRED>
```

The *search_reply*'s unique identifier.

```
<!ELEMENT search_area_list ( search_area* )>
```

A list of *search_area*s, see section 6 for a description of these elements.

```
<!ATTLIST search_area_list numberitems %number; #REQUIRED
                           total_numberitems %number; #IMPLIED
                           starting_index %number; #IMPLIED
                           ending_index %number; #IMPLIED >
```

Name	Type	Description
numberitems	integer	The number of *search_area*s in the *search_area_list*.
total_numberitems	integer	The total number of matches for the search.
starting_index	integer	The index of the first match in the *search_area_list*.
ending_index	integer	The index of the last match in the *search_area_list*.

```
<!ELEMENT search_item_list ( search_item* )>
```

A list of *search_item*s, see section 6 for a description of these elements.

```
<!ATTLIST search_item_list numberitems %number; #REQUIRED
                           total_numberitems %number; #IMPLIED
                           starting_index %number; #IMPLIED
                           ending_index %number; #IMPLIED >
```

Name	Type	Description
numberitems	integer	The number of *search_item*s in the *search_item_list*.
total_numberitems	integer	The total number of matches for the search.
starting_index	integer	The index of the first match in the *search_item_list*.
ending_index	integer	The index of the last match in the *search_item_list*.

12.3 Advertisement Debit Request

```
<!ELEMENT ad_debit (itemid) >
<!ATTLIST ad_debit type %number; #REQUIRED >

<!ELEMENT ad_debit_request (ad_debit*) >
<!ATTLIST ad_debit_request transaction_id ID #REQUIRED
                           count %number; #REQUIRED >
```

Adds an advertisement debit to the server.

Name	Type	Description
itemid	string	Unique item id.
type	integer	Type of the item.
count	integer	The number of *ad_debit* elements in the request.

12.4 Advertisement Debit Reply

```
<!ELEMENT ad_debit_reply EMPTY >
<!ATTLIST ad_debit_reply transaction_id ID #REQUIRED >
```

Reply to an ad_debit_request. Does not contain anything.

12.5 Category List Request

```
<!ELEMENT category_list_request ( position_item? ) >
<!ATTLIST category_list_request transaction_id ID #REQUIRED
                                crc %hex_t; #REQUIRED
                                language CDATA #REQUIRED >
```

Requests a category list from the server. Optionally the client may supply its position, the server will then choose a specific category list for that area (if available).

Consider to use the Local Category Tree Request instead of this one, see 12.9

Name	Type	Description
language	string	The language to translate the category names to.
crc	string	A crc in hex from a previous category list request, can be empty.

12.6 Category List Reply

```
<!ELEMENT category_list_reply (category*|crc_ok) >
<!ATTLIST category_list_reply transaction_id ID #REQUIRED
                              count %number; #REQUIRED
                              crc %hex_t; #REQUIRED >

<!ELEMENT cat (name,image_name?,cat*) >
<!ATTLIST cat cat_id %number; #REQUIRED >
```

Returns categories with their name, translated name and image filename.

Name	Type	Description
count	integer	The number of categories returned and *crc* is the crc of the entire list.
crc_ok		Returned if the *crc* matched.

12.7 Category Tree Request

```
<!ENTITY % category_tree_t "(vicinity|eventfinder)" >
<!ELEMENT category_tree_request EMPTY>
<!ATTLIST category_tree_request transaction_id ID #REQUIRED
                                crc %hex_t; #REQUIRED
                                language CDATA #REQUIRED
                                type %category_tree_t; "vicinity" >
```

Consider to use the Local Category Tree Request instead of this one, see 12.9.

Name	Type	Description
crc	string	The crc from a previous *category_tree_reply*.
language	string	The language in which the category names should be in.
type	string	Which type of tree to get. If type is vicinity then the normal poi categories will be returned. If type is eventfinder then special music event types will be returned.

12.8 Category Tree Reply

```
<!ELEMENT cat (name,image_name?,cat*) >
<!ATTLIST cat cat_id %number; #IMPLIED >
<!ELEMENT category_tree ( cat* ) >
<!ELEMENT category_tree_reply (category_tree|crc_ok|
                               ( status_code, status_message,
                                status_code_extended? )) >
<!ATTLIST category_tree_reply transaction_id ID #REQUIRED
                              crc %hex_t; #REQUIRED >
```

Name	Type	Description
cat		Node that describes a category. Can contain subnodes.
name	string	Translated name of the category.
image_name	string	Name of the image associated with the category.
cat_id	integer	Unique id of the category.
crc	string	A checksum calculated of the entire tree.

12.9 Local Category Tree Request

```
<!ELEMENT local_category_tree_request ( position_item ) >
<!ATTLIST local_category_tree_request transaction_id ID #REQUIRED
                                      crc CDATA #REQUIRED
                                      language %language_t; #REQUIRED
                                      version %number; #REQUIRED >
```

Requests a category tree from the server. The server will choose a specific category tree depending on the position provided by the client.

Name	Type	Description
crc	string	The crc from a previous *local_category_tree_reply*. Send empty attribute if no crc is available.
language	string	The language in which the category names should be in.
version	integer	The version to use in the reply.

12.10 Local Category Tree Reply

```
<!ELEMENT category_table ( #PCDATA ) >
<!ATTLIST category_table length %number; #REQUIRED >
<!ELEMENT lookup_table ( #PCDATA ) >
<!ATTLIST lookup_table length %number; #REQUIRED >
<!ELEMENT string_table ( #PCDATA ) >
<!ATTLIST string_table length %number; #REQUIRED >
<!ELEMENT local_category_tree_reply ( (category_table, lookup_table,
                                    string_table)|crc_ok|( status_code,
                                    status_message, status_uri?,
                                    status_code_extended? ) ) >
<!ATTLIST local_category_tree_reply transaction_id ID #REQUIRED
                                    crc CDATA #IMPLIED >
```

Returns the category tree in binary format (Base64 encoded). See sepcification of the format below the table.

Name	Type	Description
category_table		Node that contains the binary category table. The data is Base64 encoded.
lookup_table		Node that contains the binary lookup table. The data is Base64 encoded.
string_table		Node that contains the binary string table. The data is Base64 encoded.
length	integer	Number of items in the table. Used to improve allocation performance on client.
crc	string	A checksum calculated of the entire tree. The crc attribute is only avaiable if the request completes successfully.

Binary format specification

Version 1 of the binary format of the local category tree has the following ABNF-grammar:

```
category tree = category_table
                lookup_table
                string_table
; actually sent as 3 different entities

string_table = 1*string
string       = uint16      ; length indicator
               [1*nonnull] ; UTF8-sequence with contraints, see below.
               %x00        ; C string terminator
nonnull      = %x01-%xFF

category_table = top_level_list ; virtual root - user never sees this.
                 1*category

top_level_list = number_of_subcategories
                 0*int32      ; the sub categories as byte offsets into
                              ; category_table

category = category_id
           string_table_byte_index ; category name in used language
           string_table_byte_index ; image name as used in TMap-interface
           number_of_subcategories
           0*int32            ; the sub categories as byte offsets into
                              ; category_table

lookup_table = 1*lookup_entry    ; sorted on category_id
lookup_entry = category_id
               int32         ; byte offset of this category in category_table
```

```
category_id = int32
number_of_subcategories = uint16
string_table_byte_index = int32
```

- The list of sub categories are sorted in the order they should appear in the UI, i.e. sorted according to the rules of the language requested.
- Network byte order is used.
- Indices and offsets are absolute and never less than zero even if they have type int32 and not uint32.

The string format

- The string format is compatible both with C stdlib functions for manipulating zero-byte terminated strings and using java.io.DataInput.readUTF() to read strings on java.
- All string indexes are byte indices into the string table. The byte pointed to is the first byte after the length indicator. Thus the offset to start reading the length indicator is index-2.
- The length indicator does not count the terminating zero byte.
- The allowed Unicode code points is limited to U+0001-U+FFFF (Basic Multilingual Plane (BMP)).
- The code point U+0000 is not allowed as it would result in a 0x00-byte which would terminate the string.
- Only UTF-8 sequences that are valid and results in valid code points are allowed.

12.11 Compact Search Request

```
<!ELEMENT category_id ( #PCDATA ) >
<!ELEMENT category_query ( #PCDATA ) >
<!ELEMENT category_list ( category_id+ ) >
<!ELEMENT compact_search_request ( search_item_query,
                                   ( category_query | category_name  |
                                     category_id | category_list )?,
                                   ( (search_area_query, top_region_id)|
                                     search_area|
                                    (position_item, distance?) ) ) >
```

Name	Type	Description
category_id	integer	A unique category id, see *category_list_reply* and *category_tree_reply*
category_query	string	Match category names to this string
category_list		Contains a set of category ids to search in
search_item_query	string	Match items to this string
category_name	string	An exact name of the category
search_area_query	string	The city or area to search in
top_region_id	integer	A unique region to search in
search_area		Id of area to search in. See ref here
position_item		Coordinates for position search. See ref here
distance	integer	The radius in meters from position in *position_item*

A text search for search areas, search items or search from position.

There are three ways to search within categories. The first one is using *category_id* which is a number that comes from either the category tree or the category list.

The second way to search within categories is to do a free text category string using the *category_query* tag. This will search for categories matching this string with the language specified. The string is matched against all categories in the category tree, see *category_tree_reply*. The last way is to use an exact category name with the tag *category_name* which must contain the entire english name of the category, this search type is obsolete and must not be used in any new applications! Use the *category_id* instead.

```
<!ATTLIST compact_search_request transaction_id ID #REQUIRED
                        start_index %number; #REQUIRED
                        end_index %number; #REQUIRED
                        max_hits %number; #REQUIRED
                        language CDATA #REQUIRED
                        round %number; #REQUIRED
                        heading %number; "-1"
                        uin %number; #IMPLIED
                        version %number; "0"
                        include_category_id %bool; "false"
                        include_top_region_id %bool; "false"
                        use_persistent_ids %bool; "false"
                        position_system %position_system_t; "MC2" >
```

Name	Type	Description
transaction_id	string	Unique id for the request
start_index	integer	Start offset of the query
end_index	integer	End offset of the query
max_hits	integer	Number of maximum search results to return
language	string	Language of results
round	integer	Search round. Round 0 = Fast internal search, Round 1 = Slow external provider search
heading	integer	The heading to search in, default is all
uin	integer	Search using this user
version	integer	Version 0 is used for old clients that can not handle ad_result_text and all_result_text nodes, version 1 sends these nodes.
include_category_id	boolean	Whether to include category ids in the reply.
include_top_region_id	boolean	Whether to have country search_areas in the reply.
use_persistent_ids	boolean	Do not use this unless told to do so as turnig it on will change the ids in the results in a way that they will not work if used in a request.
position_system		Determines which coordinate system to use in the reply.

The *round* will be ignored if *heading* value is set other than *-1*.

12.12 Compact Search Reply

```
<!ELEMENT compact_search_reply (ad_results_text?,
                                all_results_text?, search_hit_list*)>
```

The text to be displayed above top hits in the heading view is controlled by *ad_results_text* value and the text bove the other headings is controlled by *all_results_text* value.

```
<!ATTLIST compact_search_reply transaction_id ID #REQUIRED>
```

transaction_id is the unique identifier of the *compact_search_reply*.

```
<!ELEMENT search_hit_list ( ad_results_text?, all_results_text?,
                            ( search_item* | search_area* ))>
```

Name	Type	Description
ad_results_text	string	The title string to display above advertisement hits inside the heading
all_results_text	string	The title string to display above normal hits (i.e non-advertisement hits) inside the heading
search_item		See search_item(6)
search_area		See search_area(6)

A list of *search_items*, see section 6 for a description of these elements. The *search_item* also includes an extra attribute *image* which is the image name, without file extension. The image can be fetched with TMap from the XML server. When *search_area* nodes are encountered then these can not be used for *poi_info_request*, instead they must be used for a new search with the old *what* field and the results *search_area* to get a list of search hits within the area.

```
<!ATTLIST search_hit_list numberitems %number; #REQUIRED
                          total_numberitems %number; #REQUIRED
                          starting_index %number; #REQUIRED
                          ending_index %number; #REQUIRED
                          heading %number; #REQUIRED
                          top_hits %number; "0" >
```

Name	Type	Description
numberitems	integer	The number of *search_items* in the *search_hit_list*.
total_numberitems	integer	The total number of matches for the search.
starting_index	integer	The index of the first match in the *search_hit_list*.
ending_index	integer	The index of the last match in the *search_hit_list*.
heading	integer	The heading identifier of the list, see *search_desc_reply*.
top_hits	integer	The number of top hits in this list. The top hits are the first hits that should be shown above all the headings.

12.13 Search Description Request

```
<!ELEMENT search_desc_request EMPTY />

<!ATTLIST search_desc_request transaction_id ID #REQUIRED
                              crc %hex_t; #REQUIRED
```

```
                        language CDATA #REQUIRED
                        uin %number; #IMPLIED
                        desc_version %number; "0" >
```

Name	Type	Description
transaction_id	integer	A unique identifier for the request.
crc	string	The previous crc from search_desc_reply.
language	string	The language preference for the reply.
uin	integer	Anothers users uin to get search_desc_request for.
desc_version	integer	Determines the descriptor version. Version 0 is for old clients (java version 7) that did not implement search result correctly and thus can not handle special headings such as phonebook and favorites. So use version 1 for new clients.

12.14 Search Description Reply

```
<!ELEMENT search_desc_reply (search_hit_type* | crc_ok ) >
```

Name	Type	Description
search_hit_type		The description of the *search_hit_list* type.
crc_ok		An empty node to indicate that the crc matched.

```
<!ATTLIST search_desc_reply transaction_id ID #REQUIRED
                            crc %hex_t; #REQUIRED
                            length %number; #REQUIRED >
```

Name	Type	Description
transaction_id	string	Unique identifier for the request.
crc	string	The check sum. The reply will be a single element *crc_ok* if crc from request matches.
length	integer	The number of *search_hit_type* in the list.

```
<!ELEMENT image_name ( #PCDATA ) >
<!ELEMENT search_hit_type (name, top_region_id?, image_name?, type?) >
```

Name	Type	Description
name	string	The translated name of the type.
top_region_id	integer	The top region id of the search hit.
image_name	string	The image name without file extension.
type	string	The type of service, localized (e.g. Yellow pages).

```
<!ATTLIST search_hit_type round %number; #REQUIRED
                          heading %number; #REQUIRED >
```

Name	Type	Description
round	integer	The round for the specific type.
heading	integer	The heading number for the type.

12.15 Search Position Description Request

```
<!ELEMENT search_position_request ( position_item ) />
```

Name	Type	Description
position_item		The position to determine top region and which search providers that are present.

This request will determine the top region and which search providers that are present at a position.

12.16 Search Position Description Reply

```
<!ATTLIST search_position_desc_request
          transaction_id ID #REQUIRED
          language CDATA #REQUIRED >
```

Name	Type	Description
language	string	The language for which the top region name and search providers will be translated to.

```
<!ELEMENT search_position_desc_reply ( top_region?, search_hit_type* ) >
<!ATTLIST search_position_desc_reply transaction_id ID #REQUIRED
                                    length %number; #REQUIRED >
```

See *top_region* and *search_hit_type*. Position system in the top region element will have the same system as the *position_item* in the request. *length* is the number of *search_hit_type* nodes, i.e excluding the top region node.

12.17 POI Search Request

```
<!ELEMENT poi_search_request (position_item, distance,
                              category_list?,
                              search_item_query?) >

<!ATTLIST poi_search_request transaction_id ID #REQUIRED
                             start_index %number; #REQUIRED
                             end_index %number; #REQUIRED
                             language CDATA #REQUIRED
                             include_top_region_id %bool; "false"
                             use_persistent_ids %bool; "false" >
```

Name	Type	Description
search_item_query	string	Match search hit names with this string.
category_list		Category ids to search in.
position_item		Centrum point for search.
distance		Radius in meters.
start_index	integer	Start offset of search hits.
end_index	integer	End offset of search hits.
language	string	The language for withc the search hits should be translated to.
include_top_region_id	boolean	Whether to have country search_areas in the reply.
use_persistent_ids	boolean	Do not use this unless told to do so as turnig it on will change the ids in the results in a way that they will not work if used in a request.

This request will search for POIs within a circle centred at *position_item* with a *distance* radius in meters. The result is a *poi_search_reply*. Note that the reply may return a different *end_index* than requested. The maximum radius is set to 100 km. The different poi types are visible at different radius ranges. Here are the ranges:

Range (in km)	POI
0-100	airport
0-20	parking, petrol stations, rent a car, hospital, hotel
0-4	All pois not covered by the above ranges.

(The ranges are closed, i.e 0 <= x <= 20)

The only poi type you can not search for is city centres.

12.18 POI Search Reply

```
<!ATTLIST poi_search_reply transaction_id ID #REQUIRED >
<!ELEMENT poi_search_reply ( search_item_list |
                             ( status_code, status_message,
                             status_uri?, status_code_extended? ) )>
```

13 Copyright Strings

Request to get the copyright strings for the map data. Contains boundingboxes for the different copyright strings.

13.1 Copyright Strings Request

```
<!ELEMENT copyright_strings_request EMPTY >
<!ATTLIST copyright_strings_request transaction_id ID #REQUIRED
                                    crc %hex_t; #REQUIRED
                                    language CDATA #REQUIRED >
```

Requests copyright strings from the server with specified language.

Name	Type	Description
crc	string	A checksum from a previous request, can be empty on the first request.
language	string	Language for the copyrights.

13.2 Copyright Strings Reply

```
<!ELEMENT copyright_strings_reply ( crc_ok | copyright_strings_data ) >
<!ATTLIST copyright_strings_reply transaction_id ID #REQUIRED
                                  crc %hex_t; #REQUIRED >
<!ELEMENT copyright_strings_data ( #PCDATA )>
```

Name	Type	Description
copyright_strings_data	string	base64 encoded format string of the copyright data
crc	string	Will be matched against the current copyright data.
crc_ok		Will be returned instead of *copyright_strings_data*. if the crc matched.

14 Expand

14.1 Expand Request

```
<!ELEMENT expand_request ( expand_request_header,
                           expand_request_query )>
```

A request to expand a category to a list of *search_item*s. Can also expand a *search_item* or a *search_area* and return it's boundingbox. Also, a *position_item* can be expanded to a *search_item*, this is, the street that is closest to the given position.

```
<!ATTLIST expand_request transaction_id ID #REQUIRED>
```

The unique identifier of the *expand_request*.

```
<!ELEMENT expand_request_header (search_preferences)>
```

The settings to use when expanding a *search_item*.

```
<!ENTITY % expand_request_location_t "(all_possible|country_city)">
<!ATTLIST expand_request_header
                        position_system %position_system_t; "MC2"
                        include_top_region_id %bool; "false"
                        location_name %expand_request_location_t; #IMPLIED  >
```

The *position_system* to use in the reply. The *include_top_region_id* determines whether the node *top_region_id* is included in the country *search_area* node. The *location_name* sets the content of any *location_name* node in the reply.

```
<!ELEMENT expand_request_query ( (search_area, search_item) |
                                 search_item | position_item | search_area )>
```

If the query contains a *search_item* to expand and that item is a category then the *search_area* in which to expand must be present to set the area from which to get points-of-interest items. If the query is a single *search_item* or *search_area* then the *boundingbox* of that item is returned. If the query is a *position_item*, the closest point on a street is returned in the form of a *search_item*.

14.2 Expand Reply

```
<!ELEMENT expand_reply ( (search_item_list | search_area_list |
                          companydata)+ |
                         (status_code, status_message,
                          status_code_extended? ) )>
```

The result of an *expand_request* with the expanded data.

```
<!ATTLIST expand_reply transaction_id ID #REQUIRED>
```

The unique identifier of the *expand_reply*.

```
<!ELEMENT companydata EMPTY>
```

To be defined. This could contain, for example, phone numbers, web address, etc. See POI info request.

15 Send SMS

15.1 Send SMS Request

```
<!ELEMENT send_sms_request ( phone_number,
                             (smsmessage |
                              (route_sms_message, route_message_data) |
                              (local_map_sms_settings, local_map_data) |
                              (wayfinder_destination_sms) |
                              (wayfinder_route_sms) |
                              (wayfinder_favourite_sms) |
                              (wap_push_service_indication) ) ) >
<!ATTLIST send_sms_request transaction_id ID #REQUIRED
                           wayfinder_sms_version CDATA "1">

A request to send one or several SMSes.


<!ELEMENT smsmessage (#PCDATA)>

Plain text to send as an SMS. The text will be truncated if it is too
long for one SMS.


<!ELEMENT route_sms_message ( phone_manufacturer, phone_model ) >
<!ATTLIST route_sms_message wap_link %bool; #REQUIRED >

Data about sender of route message such as phone model and manufacturer.
The wap_link attribute tells if the route should be sent as an SMS with
a WAP link or as SMSes with driving instructions.  If the message is
too long to fit into one SMS, it is split into several SMSes, which
may be concatenated based on the capabilities of the phone_model.


<!ELEMENT wap_push_service_indication ( #PCDATA ) >
<!ATTLIST wap_push_service_indication href %HREF; #REQUIRED >

Send WAP Push SMS with Service Indication, href is the URI the WAP browser
should go to, the contents of the element is the text shown to the user.
See WAP Forum specification WAP-167-ServiceInd-20010731-a.


<!ELEMENT route_message_data ( language, signature,
                               originString, originLocationString,
                               destinationString,
                               destinationLocationString ) >
<!ATTLIST route_message_data route_id CDATA  #REQUIRED >
<!ELEMENT signature ( #PCDATA ) >
<!ELEMENT originString ( #PCDATA ) >
```

```
<!ELEMENT originLocationString ( #PCDATA ) >
<!ELEMENT destinationString ( #PCDATA ) >
<!ELEMENT destinationLocationString ( #PCDATA ) >
```

Data and settings about the route to send in a message.

Name	Type	Description
route_id	string	The id of the route.
signature	string	A text that is placed last in the message.
originString	string	A text describing the origin of the route, such as the name of the origin.
originLocationString	string	A text describing the origin's location, such as the name of the origin city.
destinationString	string	A text describing the destination of the route.
destinationLocationString	string	A text describing the destination's location.

```
<!ELEMENT local_map_sms_settings ( phone_manufacturer, phone_model ) >
```

The settings needed to send a local map sms, i.e., an URL to a local map. See section 20.1 for a description of the *local_map_data* element.

```
<!ELEMENT wayfinder_destination_sms ( position_item, signature? ) >
<!ATTLIST wayfinder_destination_sms description CDATA #REQUIRED >
```

A Wayfinder destination sms to send to a cellular with Wayfinder program installed. The description text will be truncated to fit into a single sms.

```
<!ELEMENT wayfinder_route_sms ( position_item, position_item, signature? ) >
<!ATTLIST wayfinder_route_sms orig_description CDATA #REQUIRED
                              dest_description CDATA #REQUIRED >
```

A Wayfinder route sms to send to a cellular with Wayfinder program installed. The description text will be truncated to fit into a single sms.

Name	Type	Description
position_item		First node is position of origin the second node is the destination.
signature	string	A signature to the message.
orig_description	string	origin description.
dest_description	string	destination description.

```
<!ELEMENT wayfinder_favourite_sms ( position_item, name, short_name,
                                    category_name, map_icon_name,
                                    signature? )>
```

```
<!ATTLIST wayfinder_favourite_sms description CDATA #REQUIRED >
<!ELEMENT short_name ( #PCDATA ) >
<!ELEMENT category_name ( #PCDATA ) >
<!ELEMENT map_icon_name ( #PCDATA ) >
```

A Wayfinder favourite sms to send to a cellular with Wayfinder program installed. The description text will be truncated to fit into a single sms.

Name	Type	Description
position_item		Position of the favorite.
name	string	Name of the favorite.
short_name	string	Short version of the name.
category_name	string	Category name for the favorite.
map_icon_name	string	Name of the map icon.
signature	string	A signature to be appended to the sms.

15.2 Send SMS Reply

```
<!ELEMENT send_sms_reply ( status_code, status_message,
                           status_code_extended? )>
<!ATTLIST send_sms_reply transaction_id ID #REQUIRED>
```

The reply to a *send_sms_request* with the result status of the sending process.

16 User Login, Verify and Logout

16.1 User Login

```
<!ELEMENT user_login_request ( user_name, user_password,
                               user_service? ) >
<!ATTLIST user_login_request
             transaction_id ID          #REQUIRED
             user_create_session %bool; "false"
             client_type         CDATA  #IMPLIED
             client_type_options CDATA  #IMPLIED >
<!ELEMENT user_name ( #PCDATA ) >
<!ELEMENT user_password ( #PCDATA ) >
```

Log in a user using a user name and a password. Optionally the intended service can be set.

Name	Type	Description
user_name	string	User name.
user_password	string	Password.
user_service	string	The intended service.
user_create_session	boolean	If true then session data is returned in the reply of a successful login request.

```
<!ELEMENT user_login_reply ( status_code, status_message,
                             status_code_extended?,
                             user_session_id?, user_session_key? ) >
<!ATTLIST user_login_reply transaction_id ID #REQUIRED>
<!ELEMENT user_session_id ( #PCDATA ) >
<!ELEMENT user_session_key ( #PCDATA ) >
```

The reply to a *user_login_request* with the status of the login and optionally a session that can be used in *user_verify_requests*.

16.2 User Verify

```
<!ELEMENT user_verify_request ( user_session_id, user_session_key ) >
<!ATTLIST user_verify_request transaction_id ID #REQUIRED>
```

Request for verifying a session from a *user_login_reply*.

```
<!ELEMENT user_verify_reply ( status_code, status_message,
                              status_code_extended? ) >
<!ATTLIST user_verify_reply transaction_id ID #REQUIRED>
```

The status of a *user_verify_request*.

16.3 User Logout

```
<!ELEMENT user_logout_request ( user_session_id, user_session_key ) >
<!ATTLIST user_logout_request transaction_id ID #REQUIRED>
```

Request for ending a session.

```
<!ELEMENT user_logout_reply ( status_code, status_message,
                              status_code_extended? ) >
<!ATTLIST user_logout_reply transaction_id ID #REQUIRED>
```

The status of a *user_logout_request*.

17 Map

17.1 Map Request

```
<!ELEMENT map_request ( map_request_header, map_symbol_list? ) >
<!ATTLIST map_request transaction_id ID #REQUIRED>
<!ELEMENT map_request_header ( boundingbox, image_settings?,
                               route_data?, phone_position? ) >
<!ATTLIST map_request_header
            image_width  %number; "400"
```

```
              image_height %number; "400"
              image_default_format %route_image_format_t; "png"
              image_display_type %image_display_type; "std"
              showMap            %bool;   "true"
              showTopographMap   %bool;   "true"
              showPOI            %bool;   "true"
              showRoute          %bool;   "true"
              showScale          %bool;   "false"
              showTraffic        %bool;   "false" >
<!ELEMENT route_data ( route_id, route_turn? ) >
<!ELEMENT route_id ( #PCDATA ) >
<!ELEMENT route_turn ( #PCDATA ) >
<!ELEMENT map_symbol_list ( map_symbol_item+ ) >
<!ELEMENT map_symbol_item ( position_item, name ) >
<!ATTLIST map_symbol_item href %HREF; #REQUIRED>
```

A request for a map image. Optionally a route and turn and/or phone position can be added to the map. The optional route is specified in a *route_data* element by the *route_id* from a previous *route_reply*. The optional phone position is specified with a *phone_position* element.

If you wish to specify an area using a center point of a circle and its' radius, you can calculate the enclosing bounding box of the circle and use that boundingbox in a *map_request*.

17.2 Map Reply

```
<!ELEMENT map_reply ( href | ( status_code, status_message,
                               status_code_extended? ) ) >
<!ATTLIST map_reply transaction_id ID #REQUIRED>
<!ELEMENT href ( #PCDATA )>
```

The reply to a map request with either a URI to the map image or a status message describing the reason why it isn't a map URI.

It is true that the format of the URI is more or less straight forward to understand. However, we recommend using the XML API for posting requests for maps. The XML server then assembles a correct URL for the current format used by the HTTP server. We recommend that the URL format is not used directly to assemble custom requests. The URL format is subject to change without notice at any time, whereas the XML API is not. We thus discourage the use of the URL map request format directly without the use of the XML server.

18 Point of Interest

18.1 POI Info Request

```
<!ELEMENT poi_info_request ( search_item, language ) >
<!ATTLIST poi_info_request transaction_id ID #REQUIRED
```

```
                              position_system %position_system_t; "MC2"
                              include_category_id %bool; "false"
                              include_full_search_item %bool; "false""
                              use_persistent_ids %bool; "false" >
```

Request for information about a specific pointofinterest. The *position_system* is the desired coordinate format in the reply. If *include_category_id* is set to true the reply will contain the *category_list* node. If *include_full_search_* is set to true the reply will contain the *search_item* node and *heading* attribute. *use_persistent_ids* Do not use this unless told to do so as turnig it on will change the ids in the results in a way that they will not work if used in a request.

18.2 POI Info Reply

```
<!ELEMENT poi_info_reply ( info_item* | ( status_code, status_message,
                                          status_code_extended? ) ) >
<!ATTLIST poi_info_reply transaction_id ID #REQUIRED>
<!ELEMENT info_item ( typeName, itemName, lat?, lon?, category_list?,
                      info_field*, search_item? )>
<!ATTLIST info_item numberfields %number; #REQUIRED
                    heading %number; #IMPLIED >
<!ELEMENT typeName ( #PCDATA )>
<!ELEMENT itemName ( #PCDATA )>
<!ELEMENT info_field ( fieldName, fieldValue ) >
<!ATTLIST info_field info_type %poi_info_t; #IMPLIED >
<!ELEMENT fieldName ( #PCDATA )>
<!ELEMENT fieldValue ( #PCDATA )>
```

The reply to a *poi_info_request* with the information about the POI(s) requested.

Name	Type	Description
typeName	string	The name of the type of item, such as, “Petrol Station”.
itemName	string	The name of the item.
info_field		A list of pairs (*info_field* elements) giving additional information. One pair consists of a *fieldName* and a *fieldValue*. The type of field is in the *info_type* attribute.

19 Simple POI Description

19.1 Simple POI Description Request

```
<!ELEMENT simple_poi_desc_request EMPTY >
<!ATTLIST simple_poi_desc_request transaction_id ID #REQUIRED
                                  crc %hex_t; #REQUIRED >
```

Request for simple poi description.

19.2 Simple POI Description Reply

```
<!ELEMENT simple_poi_desc_reply ( simple_poi_desc_data | crc_ok ) >
<!ATTLIST simple_poi_desc_reply transaction_id ID #REQUIRED
                                crc %hex_t; #REQUIRED  >
```

Reply to *simple_poi_desc_request*. Will reply with *crc_ok* if crc matches else it will reply with encoded data and crc.

```
<!ELEMENT simple_poi_desc_data (#PCDATA) >
```

Contains encoded *simple_poi_desc* data. The encoding is determined by the *te* variable.

```
<!ATTLIST simple_poi_desc_data te %te_t; #REQUIRED >
```

The *te* describes the encoding of the data.

19.3 CRC OK

```
<!ELEMENT crc_ok EMPTY >
```

A tag to indicate that crc for request matches reply.

20 E-mail

20.1 E-mail Request

```
<!ELEMENT email_request ( email_request_header,
                          (route_message_data | local_map_data) ) >
<!ATTLIST email_request transaction_id ID #REQUIRED>
<!ELEMENT email_request_header ( email_address, subject,
                                 return_email_address? ) >
<!ATTLIST email_request_header
                           image_format %route_image_format_t; "png"
                           message_type %message_t; "html"
                           route_turn_image_type %route_turn_image_t; "map"
                           max_message_size %size_t; "inf"
                           overview_image_width    %size_t; #IMPLIED
                           overview_image_height   %size_t; #IMPLIED
                           route_turn_image_width  %size_t; #IMPLIED
                           route_turn_image_height %size_t; #IMPLIED
                           abbreviate_route_names  %bool;   #IMPLIED
                           route_landmarks         %bool;   #IMPLIED
                           route_only_overview     %bool;   #IMPLIED
                           invite_email            %bool;   #IMPLIED >
<!ELEMENT email_address ( #PCDATA )>
<!ELEMENT subject ( #PCDATA )>
<!ELEMENT return_email_address ( #PCDATA )>

<!ELEMENT local_map_data ( language, signature, boundingbox,
```

```
                            local_map_string, map_symbol_list ) >
<!ELEMENT local_map_string ( #PCDATA ) >
```

A request for sending an HTML, WML or SMIL email containing a route description or a local map.

Name	Type	Description
email_address	string	The address to send the e-mail to.
subject	string	The subject of the e-mail.
return_email_address	string	An optional return e-mail address.
message_type		Selects the type of markup language to use in the email, HTML, WML and SMIL is supported and HTML is default.
route_turn_image_type		Selects the type of images to show route turn with either graphical maps or symbolic pictograms. Graphical maps is the default. This attribute is only used if the message contains a route.
max_message_size		Sets the maximum size of a message in bytes. The message is split into a number of messages to fit this limit. The default value *inf*, infinity, means that there is no limit to the size of the message. The smallest *max_message_size* is 30000 bytes.
overview_image_width	integer	Route overview and local map image width. Default value depends on *message_type*
overview_image_height	integer	Route overview and local map image height. Default value depends on *message_type*.
route_turn_image_width	integer	Route turn image width. Default value depends on *message_type*.
route_turn_image_height	integer	Route turn image height. Default value depends on *message_type*.
abbreviate_route_names		Sets if street names in the route description should be abbreviated. Default is on.
route_landmarks		Sets if landmarks should be in the route description. Default is on.
route_only_overview		Sets if only an overview image should be in the route description, no pictograms and no map images. Default is off.

20.2 E-mail Reply

```
<!ELEMENT email_reply ( status_code, status_message,
```

```
                          status_code_extended? ) >
<!ATTLIST email_reply transaction_id ID #REQUIRED>
```

The reply to an *email_request* with the status of the request.

21 SMS Format

21.1 SMS Format Request

```
<!ENTITY % sms_version_t "(vicinity|eventfinder)">
<!ELEMENT invite_sms (name) >
<!ATTLIST invite_sms type %sms_version_t; #REQUIRED >
<!ELEMENT place_sms (position_item|(search_item,language)) >
<!ATTLIST place_sms type %sms_version_t; #REQUIRED >
<!ELEMENT sms_format_request ( (smsmessage, phone_manufacturer, phone_model) |
                               (route_sms_message, route_message_data) |
                               (wayfinder_destination_sms) |
                               (wayfinder_route_sms) |
                               (wayfinder_favourite_sms) | invite_sms | place_sms ) >
<!ATTLIST sms_format_request transaction_id ID #REQUIRED
                             wayfinder_sms_version CDATA "1">
```

A request for formatting a route description or any text for SMS. This includes splitting the message into several SMSes. The splitting uses SMS concatenation depending on whether the *phone_model* supports it. This request can also make a *Wayfinder* destination SMS. If element *invite_sms* exist, then an invite sms will be created and returned for the specific invite type. If element *place_sms* exist, then a place sms will be create and returned for the specific place type and position. The *search_item* element is only valid for *gigfinder type*.

21.2 SMS Format Reply

```
<!ELEMENT sms_format_reply ( (status_code, status_message,
                              status_code_extended?) |
                             sms_list )>
<!ATTLIST sms_format_reply transaction_id ID #REQUIRED>
<!ELEMENT sms_list ( smsmessage )>
```

The reply to an *sms_format_request* with the formatted SMSes or a status message describing the error.

22 Sort by Distance

22.1 Sort Dist Request

```
<!ELEMENT sort_dist_request ( (position_item | search_item),
                              (routeable_item_list | all_favorites) ) >
<!ATTLIST sort_dist_request
```

```
                        transaction_id ID #REQUIRED
                        max_number_reply_items %number; "1"
                        sort_distance %sort_distance_t; "radius"
                        route_cost %route_cost_t; "time"
                        position_system %position_system_t; "MC2"
                        route_vehicle %route_vehicle_t; "passengercar" >
<!ELEMENT all_favorites ( (user_id |
                            (user_session_id, user_session_key) ) )>
```

A request for sort a set of routeable items or all favorites for a user by their distance from an origin.

Name	Type	Description
position_item search_item		The origin to count distances from.
routeable_item_list all_favorites		The destination(s) to sort. Either a list of routeable items or all favorites for a user.
max_number_reply_items	integer	The number of sorted *sort_dist_item*s in the reply. The number of items in the reply is never larger than this value or the number of items to sort.
sort_distance		The type of distance to sort by, see entity definition Section 5.
route_cost		Used when *sort_distance* is *route* to determine how to sort routes.
position_system		Determines the coordinate system to use in the reply.
route_vehicle		Used when *sort_distance* is *route* to select the vehicle type.

22.2 Sort Dist Reply

```
<!ELEMENT sort_dist_reply ( (sort_dist_list) |
                            ( status_code, status_message,
                              status_code_extended? ) ) >
<!ATTLIST sort_dist_reply transaction_id ID #REQUIRED>
<!ELEMENT sort_dist_list ( sort_dist_item+ )>
<!ELEMENT sort_dist_item ( (position_item | search_item | favorite) ) >
<!ATTLIST sort_dist_item
                      distance %number; #REQUIRED
                      estimated_time %number; #IMPLIED>
```

The reply to an *sort_dist_request* with the sorted items.

Name	Type	Description
sort_dist_list		The list of sorted *sort_dist_item*s.
sort_dist_item		A sorted item containing the distance and the corresponding item from the request.

23 Top Region

23.1 Top Region Request

```
<!ELEMENT top_region_request ( top_region_request_header ) >
<!ATTLIST top_region_request transaction_id ID #REQUIRED >
<!ELEMENT top_region_request_header ( language ) >
<!ATTLIST top_region_request_header
                    position_system %position_system_t; "MC2"
                    country             %bool; "true"
                    state               %bool; "false"
                    internationalRegion %bool; "false"
                    metaregion          %bool; "false" >
```

A request for getting all of the top regions. The *top_region_request_header* contains the prefered language of the top regions and the coordiante system to show boundingboxes in.

Name	Type	Description
country	boolean	Should countries be returned in reply.
state	boolean	Should states be returned in reply.
internationalRegion	boolean	Should international regions be returned in reply.
metaregion	boolean	Should meta regions be returned in reply.

23.2 Top Region Reply

```
<!ELEMENT top_region_reply ( top_region_list |
                             ( status_code, status_message,
                               status_code_extended? ) )>
<!ATTLIST top_region_reply transaction_id ID #REQUIRED>
<!ELEMENT top_region_list ( top_region* )>
<!ATTLIST top_region_list numberitems %number; #REQUIRED>
```

The reply to an *top_region_request* with the top regions. If an error occurred then a *status_code* and *status_message* is returned. A top region can be used in the *search_request* to select in which top region to search.

Name	Type	Description
top_region_list		The list of *top_region*s.

24 Zoom Settings

Lists zoom settings in the server.

24.1 Zoom Settings Request

```
<!ELEMENT zoom_settings_request EMPTY >
<!ATTLIST zoom_settings_request transaction_id ID #REQUIRED
                                crc %hex_t; #REQUIRED
                                pixel_size %number; #IMPLIED >
```

Requests zoom settings from the server.

Name	Type	Description
crc		Hexadecimal checksum from a previous request, or empty
pixel_size	integer	Request a specific projection tile size. Valid values are 180 or 256. Default is 180.

24.2 Zoom Settings Reply

```
<!ELEMENT zoom_settings_reply (zoom_levels | crc_ok ) >
<!ATTLIST zoom_settings_reply transaction_id ID #REQUIRED >
```

The reply to an *zoom_settings_request*. If crc matches server zoom_settings crc then the reply to this request is *crc_ok* else there will be a list of *zoom_levels*.

```
<!ELEMENT zoom_levels (zoom_level+)>
<!ATTLIST zoom_levels crc %hex_t; #REQUIRED
                      nbr_zoom_levels %number; #REQUIRED
                      pixel_size %number; #REQUIRED>
```

A list of *zoom_levels* and the corresponding crc.

```
<!ELEMENT zoom_level EMPTY>
<!ATTLIST zoom_level max_x %number; #REQUIRED
                     max_y %number; #REQUIRED
                     min_x %number; #REQUIRED
                     min_y %number; #REQUIRED
                     zoom_level_nbr %number; #REQUIRED
                     zoom_j2me %bool; #IMPLIED>
```

The zoom level with level number *zoom_level_nbr* and its bounding box.

```
<!ELEMENT zoom_settings_crc_ok EMPTY >
```

Answer to a *zoom_settings_request* that has the the same crc as the server *zoom_settings*.

25 Phone manufacturer

This request is a simple query for all phone manufacturers.

25.1 Phone manufacturer Request

```
<!ELEMENT phone_manufacturer_request EMPTY >
<!ATTLIST phone_manufacturer_request transaction_id ID #REQUIRED>
```

A request for all phone manufacturers.

25.2 Phone manufacturer Reply

A reply to a phone manufacturer request with all phone manufacturers.

```
<!ELEMENT phone_manufacturer_reply ( phone_manufacturer_list |
                                     ( status_code, status_message,
                                       status_code_extended? ) ) >
<!ATTLIST phone_manufacturer_reply transaction_id ID #REQUIRED>
<!ELEMENT phone_manufacturer_list ( phone_manufacturer* )>
```

The reply to an *phone_manufacturer_request* with all phone manufacturers.
If an error occurred then a *status_code* and *status_message* is returned.

26 Phone model

This request is a simple query for all phone models.

26.1 Phone model Request

```
<!ELEMENT phone_model_request ( phone_manufacturer? ) >
<!ATTLIST phone_model_request transaction_id ID #REQUIRED>
```

A request for all phone models. Optionally for a specific phone manufacturer.

26.2 Phone model Reply

A reply to a phone model request with all phone models.

```
<!ELEMENT phone_model_reply ( phone_model_list |
                              ( status_code, status_message,
                                status_code_extended? ) ) >
<!ATTLIST phone_model_reply transaction_id ID #REQUIRED>
<!ELEMENT phone_model_list ( phone_model* )>
```

The reply to an *phone_model_request* with all phone requested models.
If an error occurred then a *status_code* and *status_message* is returned.

27 User track

This request is for tracking a person.

27.1 User track Request

```
<!ELEMENT user_track_request ( user_id | uin |
                               (user_session_id, user_session_key) ) >
<!ATTLIST user_track_request transaction_id ID #REQUIRED
                             start_time %time_t; #IMPLIED
                             end_time %time_t; #IMPLIED
                             max_nbr_tracks %size_t; "1"
                             position_system %position_system_t; "MC2" >
```

A request for getting the track entries for a user between two times. The *user_track_request* attributes are start and end time for which to show log entries, the maximum number of track items to return and the coordinate system to show coordinates in. The user can be identified by *user_id*, *uin* or by *user_session_id* and *user_session_key*.

27.2 User track Reply

```
<!ELEMENT user_track_reply ( ( user_track_item* ) |
                             ( status_code, status_message,
                               status_code_extended? ) ) >
<!ATTLIST user_track_reply transaction_id ID #REQUIRED>
<!ELEMENT user_track_item ( position_item ) >
<!ATTLIST user_track_item
                      time    %time_t;     #REQUIRED
                      dist    %number;     #IMPLIED
                      speed   %number;     #IMPLIED
                      source  CDATA        #REQUIRED >
```

The reply to an *user_track_request* with the track entries. If an error occurred then a *status_code* and *status_message* is returned. The *dist* attribute is the number of centimeters traveled from the last *user_track_item*. The *speed* attribute is in meter per second times 32. Time when Track point was made. If sent to server and < 100000000 then used as time before now.

28 User track add

This request is for adding track points to a user.

28.1 User track add Request

```
<!ELEMENT user_track_add_request ( ( user_id | uin |
                                     (user_session_id, user_session_key) ),
                                   user_track_item+ ) >
<!ATTLIST user_track_add_request transaction_id ID #REQUIRED >
```

Request for adding one or more track points to a user. The user can be identified by *user_id*, *uin* or by *user_session_id* and *user_session_key*.

28.2 User track add Reply

Reply to a user track add request with the status of the operation.

```
<!ELEMENT user_track_add_reply ( status_code, status_message,
                                 status_code_extended? ) >
<!ATTLIST user_track_add_reply transaction_id ID #REQUIRED >
```

The reply to an *user_track_add_request* with the status of the operation in *status_code* and *status_message*.

29 User debit log

This request is for showing a users debit log.

29.1 User debit log request

```
<!ELEMENT user_debit_log_request ( user_id |
                                   (user_session_id, user_session_key) ) >
<!ATTLIST user_debit_log_request
                                   transaction_id ID       #REQUIRED
                                   start_time     %time_t; #REQUIRED
                                   end_time       %time_t; #REQUIRED
                                   start_index    %size_t; "0"
                                   end_index      %size_t; "99" >
```

A request for showing the debit log for a user between two times. The *user_debit_log_request* attributes are start and end time for which to show log entries, the start and end index among the log entries. The user can be identified by *user_id* or by *user_session_id* and *user_session_key*.

29.2 User debit log reply

```
<!ELEMENT user_debit_log_reply ( ( user_debit_log_element* ) |
                                   ( status_code, status_message,
                                     status_code_extended? ) ) >
<!ATTLIST user_debit_log_reply transaction_id ID       #REQUIRED
                               start_index    %size_t; #REQUIRED
                               end_index      %size_t; #REQUIRED
                               total_number_elements %size_t; #REQUIRED >
<!ELEMENT user_debit_log_element EMPTY >
<!ATTLIST user_debit_log_element
                                 message_id    %number; #REQUIRED
                                 debit_info    %number; #REQUIRED
                                 time          %time_t; #REQUIRED
                                 operationType %number; #REQUIRED
                                 sentSize      %size_t; #REQUIRED
                                 userOrigin    CDATA    #REQUIRED
                                 serverID      CDATA    #REQUIRED
                                 description   CDATA    #REQUIRED >
```

The reply to an *user_debit_log_request* with the debit log entries. If an error occurred then a *status_code* and *status_message* is returned.

30 User find

This request is for finding users from a set of search parameters.

30.1 User find request

```
<!ELEMENT user_find_request ( user ) >
<!ATTLIST user_find_request transaction_id ID #REQUIRED >
```

A request for finding users matching the values sent in the user element. The values in the user element is the ones that the user should have to match. The password is not matched. In a user's phone element only the phone_number is used when matching.

30.2 User find reply

The reply contains the user_ids of the users that match the request.

```
<!ELEMENT user_find_reply ( (user_id,uin)* |
                            (user_session_id, user_session_key) ) >
<!ATTLIST user_find_reply transaction_id ID #REQUIRED >
```

The reply to an *user_find_request* with the user_id and uins of the users that match the request. If an error occurred then a *status_code* and *status_message* is returned.

31 Transactions

This request is for changing and showing a user's transactions count.

31.1 Transactions request

```
<!ELEMENT transactions_request ( (user_id |
                                  (user_session_id,
                                   user_session_key))? ) >
<!ATTLIST transactions_request
                       transaction_id ID #REQUIRED
                       uin CDATA #IMPLIED
                       transaction_change %number; #IMPLIED >
```

The user can be identified in three different ways. First using the *user_id*, secondly using a *user_session_id* and *user_session_key* and thirdly from an *uin*.

The *transaction_change* tells the amount of transactions to add or remove if negative.

31.2 Transactions reply

The reply to an *Transactions request* with the current amount of transactions.

```
<!ELEMENT transactions_reply ( status_code, status_message,
                               status_code_extended? ) >
```

```
<!ATTLIST transactions_reply transaction_id ID #REQUIRED
                             nbr_transactions %number; #IMPLIED >
```

The reply to an *Transactions request* with the status of the operation and the current amount of transactions if ok.

32 Transaction days

This request is for changing and showing a user's transaction days.

32.1 Transaction days request

```
<!ELEMENT transaction_days_request (
   (user_id | (user_session_id, user_session_key))? ) >
<!ATTLIST transaction_days_request
                      transaction_id     ID       #REQUIRED
                      uin                CDATA    #IMPLIED
                      check              %bool;   #IMPLIED
                      transaction_change %number; #IMPLIED >
```

The user can be identified in three different ways. First using the *user_id*, secondly using a *user_session_id* and *user_session_key* and thirdly from an *uin*.

The *transaction_change* tells the amount of transaction days to add or remove if negative. The *check* says if to check if a new transaction day is needed and start it by setting a new *current_day* and decrease number of transaction days left.

32.2 Transaction days reply

The reply to an *Transaction days request* with the current amount of transaction days left and the start of *current_day*.

```
<!ELEMENT transaction_days_reply ( status_code, status_message,
                                   status_code_extended? ) >
<!ATTLIST transaction_days_reply
                      transaction_id       ID       #REQUIRED
                      nbr_transaction_days %number; #REQUIRED
                      current_day          %time_t; #REQUIRED >
```

The reply to an *Transaction days request* with the status of the operation and the current amount of transaction days left and *current_day* if ok.

33 Activation

This request is for using an activation code.

33.1 Activation request

```
<!ELEMENT activate_request ( phone_number?, new_password?,
```

```
                              name?, email?, opt_in?,
                              (external_auth | server_auth_bob |
                               handle_me | hardware_id | hardware_key+ )? ) >
<!ATTLIST activate_request      transaction_id     ID       #REQUIRED
                                activation_code    CDATA    #IMPLIED
                                uin                CDATA    #IMPLIED
                                may_use            %bool;   "true"
                                create_new_token   %bool;   "true"
                                top_region_id      %number; #IMPLIED >
<!ELEMENT email ( #PCDATA ) >
<!ELEMENT opt_in EMPTY>
<!ATTLIST opt_in name CDATA #REQUIRED>
<!ELEMENT external_auth EMPTY>
<!ATTLIST external_auth type CDATA #REQUIRED>
<!ELEMENT handle_me ( licence_key? ) >
<!ATTLIST handle_me >
<!ELEMENT licence_key EMPTY>
<!ATTLIST licence_key key CDATA #REQUIRED >
<!ELEMENT hardware_id ( #PCDATA ) >
<!ATTLIST hardware_id type %hardware_key_type_t; #REQUIRED >
<!ELEMENT hardware_key ( #PCDATA ) >
<!ATTLIST hardware_key type %hardware_key_type_t; #REQUIRED >
```

Name	Type	Description
phone_number	string	Added to user if not already present and not in other user.
new_password	string	The new password for the user.
name	string	Name for the user.
email	string	Email address for the user.
opt_in		The optional thing the user has accepted, name specifies what the user has opt:ed in on.
external_auth		Used if client is authenticated via external entity.
activation_code	string	The activation code to use. If left out some other method of authenticating must be sent, such as hardware_id or external_auth.
uin	integer	The uin of the user activating. May be left out.
may_use	boolean	If the activation code may be used, default true. If *may_use* is set to false the server will not consume an unused activation code.
create_new_token		If to create a new token for the user, default true.
top_region_id	integer	The selected region if activation code needs it, default not set.
handle_me		If to let the server create a user account. Errorcode -213 if server may not create an account. This is retained for existing clients but won't be used for new ones.
hardware_id	string	A string representing some hardware unique key. There are no requirements on formatting and server may use normalized forms for storing and comparing.
hardware_key		Contains unique hardware keys identifying the device the client is on.

```
<!ENTITY % hardware_key_type_t "(imei|btmac|bbpin|imsi|esn|
                                 phone_msisdn|iphone_dev_id|
                                 customer_msisdn)">
```

The type of hardware key.

Value	Description
imei	IMEI of the terminal
btmac	Bluetooth MAC address. Used on non-blackberry devices that have hardware bluetooth but no possibility for unsigned applications to retrieve IMEI. E.g. Nokia S40v3.
bbpin	On RIM Blackberry terminals the PIN is used. This is not the user's simcard PIN but a rather a IMEI for Blackberries. Blackberries in non-GSM-networks (e.g. IDEN) do not have IMEI but they always have a unique PIN.
imsi	International Mobile Subscriber Identity.
esn	Electronic Serial Number. CDMA's equivalent of IMEI.
phone_msisdn	An MSISDN, could be entered by the user and thus may not be 100% correct.
iphone_dev_id	iPhone hardware key.
customer_msisdn	An MSISDN, ok to use for billing purposes etc.

33.2 Activate reply

The reply to an *Activate request* with the result of the activation.

The status codes that may be returned in an activate reply:

Code	Description
-301	Too short password. The new password is too short.
-302	Bad activation code. The activation code does not exist.
-303	Used activation code. The activation code is used.
-304	Wrong phone number. The phonenumber is not valid.
-305	Extension not allowed. Must have previous activation to be able to extend.
-306	Creation not allowed. May not create new user with this activation code.
-307	May not use activation code. Attribute *may_use* is false and activation code is not used.
-308	Invalid email. Not valid email address.
-309	Moving of license key prohibited. At least one of the users may not change license key.
-310	Neither uin, activation_code nor hardware_id included in request.
-311	License key owned by more than one user. The hardware_id in the request is owned by several users and we cannot determine which to use.

```
<!ELEMENT activate_reply ( ( user_id?, auth_token? ) |
                           ( status_code, status_message,
                             status_code_extended? ) ) >
<!ATTLIST activate_reply  transaction_id       ID       #REQUIRED
                          uin                 CDATA    #IMPLIED >
<!ELEMENT auth_token ( #PCDATA ) >
```

Name	Type	Description
user_id		The current user id of the user.
auth_token		The new authentication token.

34 External Services

34.1 External services request

Request to be used e.g. on client startup up to get the list of available services.

```
<!ELEMENT ext_services_request EMPTY>
<!ATTLIST ext_services_request  transaction_id     ID           #REQUIRED
                                crc                CDATA        #IMPLIED
                                language           %language_t; #REQUIRED
                                new_client         %bool; "false" >
```

The *crc* is the crc that the client received the last time it asked for the list of external services. If no list has been received and empty crc should be ent. The *language* is the language as described elsewhere in this document. It will affect the names of the services and the names of the fields returned in the reply. The *new_client* determines if the list should contain external search services which can return *search_area* hits and has more or less than two icons in the icon list.

34.2 External services reply

Reply to the *ext_services_request*. Contains a list of external services if the crc of the server list differs from the crc sent in from the client in the request. If the crc matches, a single tag *ext_service_crc_ok* will be returned.

```
<!ELEMENT ext_services_reply ( ext_services_crc_ok |  ext_service* |
                             ( status_code, status_message,
                               status_code_extended? ) ) >
<!ATTLIST ext_services_reply   crc                 CDATA       #REQUIRED
                               transaction_id      ID          #REQUIRED
                               nbr_services        %number;    #REQUIRED >
```

34.3 External services example

34.3.1 Wrong or no crc in client list

A client sends the following request.

```
<isab-mc2>
   <auth>
      <auth_user>xox</auth_user>
      <auth_passwd>xoxox</auth_passwd>
   </auth>

   <ext_services_request transaction_id="xox"
                         language="swedish" crc="" />
</isab-mc2>
```

The server replies with:

```
<isab-mc2>
   <ext_services_reply crc="E3870B7C" transaction_id="xox">
      <ext_service service_id="1" type="search">
         <name>Eniro WP Sverige</name>
         <field id="1" req="1" type="choice" nbr_choices="1">
            <field_name>Land</field_name>
            <field_option id="1">
               <field_option_name>Sverige</field_option_name>
            </field_option>
         </field>
         <field id="2" req="1" type="string">
            <field_name>Namn/Telefonnummer</field_name>
         </field>
         <field id="3" req="0" type="string">
            <field_name>Adress/Stad</field_name>
         </field>
      </ext_service>
   </ext_services_reply>
</isab-mc2>
```

The field descriptions contain the type of the fields which can be *string*, *number* or *choice*. *String* means any string, *number* means a positive integer. If the type is *choice*, the *field* element will be followed by one or more *field_option* elements which contain the name of the option and the id to send back to the server when it has been selected.

The *req* attribute of the *field* elements is used to describe when all necessary fields have been filled in. The requirements should be checked by the client before sending an incomplete request to the server. The algorithm to determine if all necessary fields are filled in is as follows:

```
   ok = all bits set to 1
   foreach field with req != 0:
      if field filled in:
         ok = ok & field.req
      else:
         ok = ok & ^(field.req)
    if ok != 0:
       send_to_server
```

34.3.2 Correct crc in client list

A client sends the following request.

```
<isab-mc2>
   <auth>
      <auth_user>xox</auth_user>
      <auth_passwd>xoxox</auth_passwd>
   </auth>

   <ext_services_request transaction_id="xox2"
                         language="swedish" crc="E3870B7C" />
</isab-mc2>
```

The server replies with:

```
<isab-mc2>
   <ext_services_reply crc="E3870B7C" transaction_id="xox2">
      <ext_services_crc_ok/>
   </ext_services_reply>
</isab-mc2>
```

34.4 External search request

This is an example of an *external_search_request*, see 46 for definition.

```
<isab-mc2>
   <auth>
      <auth_user>w</auth_user>
      <auth_passwd>y</auth_passwd>
   </auth>
   <ext_search_request language="swedish"
                       search_item_starting_index="0"
                       search_item_ending_index="10"
                       service_id = "1"
                       transaction_id="xox4mati">
   <field_val id="1">1</field_val>
   <field_val id="2">2882200</field_val>
</ext_search_request>
</isab-mc2>
```

The reply is an ordinary search reply or an error message if all necessary fields weren't correctly filled in. See 34.2 for field information.

35 Tunnel

This request is for tunneling internet traffic through the server.

35.1 Tunnel Request

```
<!ELEMENT tunnel_request ( post_data? ) >
<!ATTLIST tunnel_request   transaction_id     ID       #REQUIRED
```

```
                          url                %HREF;   #REQUIRED  >
<!ELEMENT post_data ( #PCDATA ) >
<!ENTITY % te_t "(identity|base64)" >
<!ATTLIST post_data te %te_t; "identity">
```

Request for retreiving an URL with optional post data.

35.2 Tunnel Reply

Reply to a tunnel request with the reply or error reply.

```
<!ELEMENT tunnel_reply ( ( header*, body? ) |
                         ( status_code, status_message,
                           status_code_extended? ) ) >
<!ATTLIST tunnel_reply  transaction_id       ID       #REQUIRED
                        status_line         CDATA    #IMPLIED >
<!ELEMENT header EMPTY>
<!ATTLIST header field CDATA #REQUIRED
                 value CDATA #REQUIRED >
<!ELEMENT body ( #PCDATA ) >
<!ATTLIST body te %te_t; "base64">
```

The reply to a *tunnel_request* with reply.

36 POI Review

The POI Review interfaces enables users to share their thoughts on POIs delivered through the Wayfinder service. Each user may review and grade each POI once. The user may change their review at a later time. Users may see reviews posted by other users, but not change them.

36.1 POI Review Request

```
<!ELEMENT poi_review_requests ( poi_review_add_request |
                                poi_review_delete_request |
                                poi_review_list_request )+ >
```

The POI review request is a container for more specific POI review actions. Each of the actions is presented in some detail below.

36.2 POI Review Reply

```
<!ELEMENT poi_review_replies ( poi_review_add_reply |
                               poi_review_delete_reply |
                               poi_review_list_reply )+ >
```

The POI Review Reply is, like the POI Review Request, a container of more specific replies to POI review actions. Each POI review action must be answered by a POI review action reply, identified by the proper transaction id.

Note that if the request document contained several *poi_review_request* elements, the server does not guarantee that the individual replies are sent in corresponding *poi_review_reply* groups.

36.3 POI Review Common Elements

```
<!ENTITY % poi_grade "(0|1|2|3|4|5)" >
```

The user may grade the POI while submitting his review. The grade is a value from 1 to 5 inclusive where 1 is the lowest and 5 the highest.

```
<!ELEMENT poi_review_title ( #PCDATA ) >
```

Each review must have a title or abstract that can be presented in list format.

```
<!ELEMENT poi_review_text ( #PCDATA ) >
```

The actual review text should be submitted in this element.

36.4 POI Review Add Request

```
<!ELEMENT poi_review_add_request ( ( user_id | uin |
                                     ( user_session_id, user_session_key ) ) ,
                                   ( poi_review_title, poi_review_text )? ) >
<!ATTLIST poi_review_add_request transaction_id  ID           #REQUIRED
                                 poi_id          CDATA        #REQUIRED
                                 grade           %poi_grade;  #REQUIRED
                                 lang            %language_t; #IMPLIED
                                 review_id       CDATA        #IMPLIED >
```

User reviews are added using the *poi_review_add_request* element.

The *user_id*, *uin*, or *user_session_id* and *user_session_key* are used to identify the user that is submitting this review.

If the user just wants to grade a POI, not write a review, the *poi_review_title* and *poi_review_text* elements may be excluded.

Name	Type	Description
poi_review_title	string	Shall hold the title or caption of the request. Note that the title should not be too long.
poi_review_text	string	Shall hold the review text.
transaction_id	string	Arbitrary text string to identify this request.
poi_id	string	Shall hold a string that uniquely identifies the POI that this review is for.
grade		The number of ’stars’ the user wants to give this POI.
lang	string	The language of the review.
review_id		The id of the review to modify.

status codes:

Code	Description
-600	Not found.
-601	Not allowed.

36.5 POI Review Add Reply

```
<!ELEMENT poi_review_add_reply ( status_code, status_message,
                                 status_code_extended? )? >
<!ATTLIST poi_review_add_reply transaction_id  ID     #REQUIRED
                               review_id       CDATA  #IMPLIED >
```

If the request qould not be processed for some reason, the *status_code*, *status_message*, and *status_code_extended* will be included to tell you why.

Name	Type	Description
transaction_id	string	String that matches the *transaction_id* of the corresponding *poi_review_add_request*.
review_id	string	String that uniquely identifies this review.

36.6 POI Review Delete Request

```
<!ELEMENT poi_review_delete_request ( user_id | uin |
                                      ( user_session_id, user_session_key ) ) >
<!ATTLIST poi_review_delete_request transaction_id  ID     #REQUIRED
                                    review_id       CDATA  #REQUIRED >
```

Requests the removal of a review. The user identified by *user_id*, *uin*, or *user_session_id* and *user_session_key* must match the user listed as the author of the review.

Name	Type	Description
transaction_id	string	Arbitrary string identifier that uniquely identifies this request whithin this document.
review_id	string	String ID of the review that should be removed.

36.7 POI Review Delete Reply

```
<!ELEMENT poi_review_delete_reply ( status_code, status_message,
                                    status_code_extended? )? >
<!ATTLIST poi_review_delete_reply transaction_id ID #REQUIRED >
```

If the request qould not be processed for some reason, the *status_code*, *status_message*, and *status_code_extended* will be included to tell you why.

36.8 POI Review List Request

```
<!ELEMENT poi_review_list_request ( user_id | uin |
                                    ( user_session_id, user_session_key ) |
                                    poi_review_poi | poi_review_id ) >
<!ATTLIST poi_review_list_request transaction_id  ID                    #REQUIRED
```

```
                                  details         %poi_review_details; "all"
                                  lang            %language_t;         #IMPLIED >
<!ENTITY % poi_review_details "(none|some|all)" >
<!ELEMENT poi_review_poi EMPTY >
<!ATTLIST poi_review_poi poi_id CDATA #REQUIRED >
<!ELEMENT poi_review_id  EMPTY >
<!ATTLIST poi_review_id  review_id CDATA #REQUIRED >
```

Request a list of POI reviews. Which reviews should be listed is indicated by the contained elements.

user_id, uin, or user_session_id and user_session_key All reviews by this user.

poi_review_id All POI reviews with the id encoded in the *review_id* attribute. This should only be one review.

poi_review_poi All POI reviews for the POI identified by the *review_id* attribute.

Some aspects of the list may be controlled using atributes.

transaction_id Arbitrary string that uniqely identifies this request whithin this document.

details Specifies how much details should be included in the *poi_review_list_reply*.

none Only overview information of each reveiwed POI will be listed.

some Overview information and the user, grade, and title of each review will be listed.

all All information will be listed.

lang If the *lang* attribute is set, only reviews of this language will be included in the list.

36.9 POI Review List Reply

```
<!ELEMENT poi_review_list_reply ( ( poi_review* ) |
                                  ( status_code, status_message,
                                    status_code_extended? ) ) >
<!ATTLIST poi_review_list_reply transaction_id  ID      #REQUIRED >

<!ELEMENT poi_review ( poi_review_detail* ) >
<!ATTLIST poi_review poi_id      CDATA        #REQUIRED
                     avg_grade   CDATA        #IMPLIED
                     grade_count %number;     #IMPLIED >

<!ELEMENT poi_review_detail ( user_id, poi_review_title, poi_review_text? ) >
<!ATTLIST poi_review_detail review_id  CDATA       #REQUIRED
                            date       CDATA       #REQUIRED
                            grade      %poi_grade; #REQUIRED
                            logonID    CDATA       #IMPLIED
                            firstname  CDATA       #IMPLIED
                            lastname   CDATA       #IMPLIED >
```

If the request could not be processed for some reason, the *status_code*, *status_message*, and *status_code_extended* will be included to tell you why.

Otherwise a number of *poi_review* elements will be listed, each containing the reviews for one POI.

The *transaction_id* attribute will contain the string from the corresponding *poi_review_list_request*.

Each POI that has any review and is included in the list is represented by a *poi_review* element.

Name	Type	Description
poi_id	string	The POI ID.
avg_grade		The average grade calculated from all reviews for the POI, whether they are included in the list or not.
grade_count	integer	The number of grades that have been used to calculate the average.

If the *detail* attribute of the *poi_review_list_request* was set to “none” no *poi_review_detail* elements will be included, otherwise all reviews for the POI will be included unless filtered by the *lang* attribute.

The *poi_review_detail* elements holds information about a single review.

Name	Type	Description
user_id		The user that submitted the review.
poi_review_title		The title of the review.
poi_review_text		The review text. This element is only included if the *details* attribute of the *poi_review_list_request* was set to “all”.
review_id		The ID of the review.
date		The time when the review was submitted.
grade		The grade given to the POI in this review.
logonID		The user name of the reviewer.
firstname		The first name of the reviewer.
lastname		The last name of the reviewer.

37 Get client type information

This request is for getting information about a client type.

37.1 Get client type info Request

```
<!ELEMENT client_type_info_request EMPTY >
```

```
<!ATTLIST client_type_info_request
   transaction_id      ID           #REQUIRED
   client_type         CDATA        #REQUIRED
   client_type_options CDATA        #REQUIRED >
```

Request for retreiving data for a client type.

37.2 Get client type info Reply

Reply to a get client type info request with the reply or error reply.

```
<!ELEMENT client_type_info_reply ( (status_code, status_message,
                                    status_code_extended?)? ) >
<!ATTLIST client_type_info_reply
   transaction_id      ID           #REQUIRED
   phoneModel          CDATA        #REQUIRED
   imageExtension      CDATA        #REQUIRED
   extraRights         CDATA        #REQUIRED >
```

The reply to a *client_type_info_request* with reply.
Status codes:

Code	Description
-700	No such client type.

38 Get server list for client type

This request is for getting server list for a client type.

38.1 Get server list for client type Request

```
<!ELEMENT server_list_for_client_type_request EMPTY >
<!ATTLIST server_list_for_client_type_request
   transaction_id      ID           #REQUIRED
   client_type         CDATA        #REQUIRED
   client_type_options CDATA        #REQUIRED
   srvt                CDATA        #REQUIRED
   uin                 CDATA        #IMPLIED >
```

Request for retreiving server list for a client type.

38.2 Get server list for client type Reply

Reply to a get server list for client type request with the reply or error reply.

```
<!ELEMENT server_list_for_client_type_reply ( server_list |
                                              (status_code, status_message,
                                               status_code_extended?) ) >
<!ATTLIST server_list_for_client_type_reply
   transaction_id      ID           #REQUIRED >
```

The reply to a *server_list_for_client_type_request* with reply.
Status codes:

Code	Description
-700	No such client type.

39 Create Wayfinder User

This request is for creating Wayfinder users.

39.1 Create Wayfinder User Request

```
<!ELEMENT create_wayfinder_user_request ( hardware_key+ ) >
<!ATTLIST create_wayfinder_user_request
   transaction_id      ID           #REQUIRED
   client_type         CDATA        #REQUIRED
   client_type_options CDATA        #REQUIRED
   client_lang         %language_t; #REQUIRED
   logon               CDATA        #REQUIRED
   password            CDATA        #REQUIRED
   activation_code     CDATA        #IMPLIED
   top_region_id       %number;     #IMPLIED >
```

Request for creating a new Wayfinder user.

39.2 Create Wayfinder User Reply

Reply to a create wayfinder user request with the reply or error reply.

```
<!ELEMENT create_wayfinder_user_reply ( (status_code, status_message,
                                         status_code_extended?,
                                         server_list?)? ) >
<!ATTLIST create_wayfinder_user_reply transaction_id ID    #REQUIRED
                                      uin            CDATA #IMPLIED >
```

The reply to a *create_wayfinder_user_request* with reply.
Status codes:

Code	Description
-700	No such client type.
-211	Redirect code. The *server_list* element gives the server to use.
-302	Problem with activation code.
-306	Problem with activation code.

40 Update Hardware Keys

This request is for changing licence keys for a user.

40.1 Update Hardware Keys Request

```
<!ELEMENT update_hardware_keys_request ( hardware_key+ ) >
<!ATTLIST update_hardware_keys_request
   transaction_id      ID           #REQUIRED
   uin                 CDATA        #IMPLIED
   client_type         CDATA        #IMPLIED
   client_type_options CDATA        #IMPLIED >
```

Request for updating a user with a set of licence keys.

40.2 Update Hardware Keys Reply

Reply to a update hardware keys request with the reply or error reply.

```
<!ELEMENT update_hardware_keys_reply ( (status_code, status_message,
                                        status_code_extended?)? ) >
<!ATTLIST update_hardware_keys_reply transaction_id ID #REQUIRED >
```

The reply to a *update_hardware_keys_request* with reply.

41 Get Stored User Data

This request and reply for getting user data such as i.e. getting the last currency conversion, the last weather searches etc.

41.1 Get Stored User Data Request

The get stored user data request will get a certain data for a specific user defined by the *uin* and *key*.

```
<!ELEMENT get_stored_user_data_request EMPTY >
<!ATTLIST get_stored_user_data_request
   transaction_id       ID          #REQUIRED
   uin                  CDATA       #REQUIRED
   key                  CDATA       #REQUIRED >
```

Name	Type	Description
key	string	The key for a certain value.

41.2 Get Stored User Data Reply

Is the reply to a get stored user data request reply or a error reply.

```
<!ELEMENT get_stored_user_data_reply ( ( stored_user_data ) |
            ( status_code, status_message, status_code_extended? ) ) >
<!ATTLIST get_stored_user_data_reply
   transaction_id       ID          #REQUIRED >

<!ELEMENT stored_user_data EMPTY >
<!ATTLIST stored_user_data
   key                  CDATA       #REQUIRED
   value                CDATA       #REQUIRED >
```

Name	Type	Description
key	string	The key for a certain value.
value	string	The value of a stored data specified by a key.

Status codes:

Code	Description
-800	No such key in the database!

42 Set Stored User Data

This request and reply for setting user data such as i.e. setting the last currency conversion, the last weather searches etc.

42.1 Set Stored User Data Request

The set stored user data request will set certain data for a specific user defined by the *uin* and *stored_user_data*.

```
<!ELEMENT set_stored_user_data_request ( stored_user_data ) >
<!ATTLIST set_stored_user_data_request
   transaction_id       ID          #REQUIRED
   uin                  CDATA       #REQUIRED >
```

42.2 Set Stored User Data Reply

Is the reply to a set stored user data request reply or a error reply.

```
<!ELEMENT set_stored_user_data_reply ( ( status_code, status_message,
                                         status_code_extended? )? ) >
<!ATTLIST set_stored_user_data_reply
   transaction_id       ID          #REQUIRED >
```

43 One Search

A new search which searches all, allowed, sources and returns a single sorted list of search matches.

43.1 One Search Request

The request just has the options needed right now and will get more when they are needed.
The location where the search should be performed is specified by either sending a position_item or a query_location and a top_region_id.

```
<!ENTITY % sorting_t "(alfa_sort|distance_sort)">
<!ENTITY % search_for_type_t "(address|all)">
<!ELEMENT query_location ( #PCDATA )>
```

```
<!ELEMENT one_search_request ( search_match_query?,
                               category_list?,
                               ( ( position_item, distance? ) |
                               ( query_location, top_region_id ) ) ) >
<!ATTLIST one_search_request transaction_id    ID            #REQUIRED
                             max_number_matches %number;     #REQUIRED
                             language          %language_t;  #REQUIRED
                             round             %number;      #REQUIRED
                             version           %number;      #REQUIRED
                             include_detail_fields %bool;      #IMPLIED
                             position_system %position_system_t; "MC2"
                             sorting           %sorting_t;   #REQUIRED
                             search_type      %search_for_type_t; "all">
<!ELEMENT search_match_query ( #PCDATA )>
```

Name	Type	Description
search_match_query	string	String to match.
category_list	element	Contains a set of category ids to match.
position_item		Coordinates for proximity search.
distance	integer	The radius in meters from position in *position_item*.
query_location	element	The city or area to seach in.
top_region_id	element	A unique region to search in.
transaction_id	string	Unique identifier for the request.
max_number_matches	integer	Maximum number of search results in reply.
language	enumeration	Language used when there is a choice.
round	integer	Search round. Round 0 = Fast internal search, Round 1 = Slow external provider search.
version	integer	Version 0 is the initial version.
include_detail_fields	boolean	Whether to include info fields, default true.
position_system	enumeration	Determines which coordinate system to use in the reply.
sorting	enumeration	How the results should be sorted.
search_type	enumeration	What we are searching for. The search type "address" is intended for round 0.

43.2 One Search Reply

Is the search result for an one search request or an error reply.

```
<!ENTITY % search_match_type_t "(street|pointofinterest|misc|person|
                                 other)">
```

```
<!ELEMENT search_match ( name, itemid,
                         location_name, lat?, lon?, category_list?,
                         search_area*, detail_item? )>
```

The *search_match* is a fresh start for search replies.

Name	Type	Description
search_match_type_t	entity	The type of match.
search_match	element	Representing a search match.
name	string	Name of the *search_match*.
itemid	string	A unique id for this match. Not persistent do not store permanently.
location_name	string	String describing the location of the match.
lat	element	Latitude coordinate.
lon	element	Longitude coordinate.
category_list	element	The search categories the match belongs to.
search_area	element	The areas that this match belong to. See *search_area*.
detail_item	element	Detailed information for the match, is added if *include_detail_fields* is *true*.

```
<!ATTLIST search_match search_match_type %search_match_type_t; #REQUIRED
                       category_image CDATA #REQUIRED
                       provider_image CDATA #REQUIRED
                       brand_image CDATA #REQUIRED
                       additional_info_exists %bool; #REQUIRED >
```

Name	Type	Description
search_match_type	enumeration	The type of match.
category_image	string	The category icon.
provider_image	string	The provider icon.
brand_image	string	The brand icon.
additional_info_exists	boolean	True if more information about this search match can be fetched.

Images are without file extension and are empty if there is no such image for the *search_match*.

```
<!ELEMENT one_search_reply ( search_list |
                             ( status_code, status_message,
                               status_uri? ) ) >
<!ATTLIST one_search_reply transaction_id  ID      #REQUIRED >

<!ELEMENT search_list ( search_match* )>
<!ATTLIST search_list number_matches       %number; #REQUIRED
                      total_number_matches %number; #REQUIRED >
```

Name	Type	Description
search_list	element	The result with the matches.
transaction_id	string	Unique identifier for the request.
number_matches	integer	The number of *search_match*es in the *search_list*.
total_number_matches	integer	The total number of matches.

44 Server Info

At the moment this request only handles new versions of client types. More information may be added when needed.

44.1 Server Info Request

The request just has the client type right now and will get more when they are needed.
This request should be sent to the server at client start up.

```
<!ELEMENT server_info_request EMPTY>
<!ATTLIST server_info_request transaction_id      ID            #REQUIRED
                              client_type         CDATA         #REQUIRED
                              client_type_options CDATA         #IMPLIED
                              client_version      CDATA         #REQUIRED >
```

Name	Type	Description
client_type	string	The client type used.
client_type_options	string	Options string for the used client type.
client_version	string	The current version of the client.

44.2 Server Info Reply

Is the result for an server info request or an error reply.

```
<!ELEMENT server_info_reply ( client_type_info |
                              ( status_code, status_message,
                                status_uri? ) ) >
<!ATTLIST server_info_reply transaction_id  ID      #REQUIRED >

<!ELEMENT client_type_info EMPTY>
<!ATTLIST client_type_info  upgrade_available    %bool; #REGUIRED
                            latest_version       CDATA  #REQUIRED
                            force_upgrade        %bool: #REQUIRED
                            upgrade_id           CDATA  #IMPLIED >
```

Name	Type	Description
upgrade_available	boolean	True if a newer version exists for this client type.
latest_version	string	The latest version available for this client type. Empty string if no version information is available for the requested client type.
force_upgrade	boolean	True if upgrade to the latest version should be forced.
upgrade_id	string	String identifying the latest version in the platform market. May be an URI, a package name, an id etc.

45 Point of Interest - Details

45.1 POI Details Request

```
<!ELEMENT poi_detail_request ( itemid ) >
<!ATTLIST poi_detail_request transaction_id ID           #REQUIRED
                             language       %language_t; #REQUIRED>


Request for information about a specific point of interest.
```

45.2 POI Details Reply

```
<!ELEMENT poi_detail_reply ( ( detail_item, resources? ) |
                           ( status_code, status_message,
                             status_code_extended? ) ) >
<!ATTLIST poi_detail_reply transaction_id ID #REQUIRED>
<!ELEMENT detail_item ( detail_field* )>
<!ATTLIST detail_item numberfields %number; #REQUIRED >

<!ELEMENT detail_field ( fieldName, fieldValue ) >
<!ATTLIST detail_field detail_type %poi_detail_t; #IMPLIED
                       detail_content %poi_detail_content_t; #IMPLIED >

<!ENTITY % poi_detail_t "(dont_show|text|street_address|full_address|phone_number|
                          url|email|poi_url|poi_thumb|average_rating|
                          description|open_hours|provider_info)" >
<!ENTITY % poi_detail_content_t "(text|phone_number|url|email_address|
                                  integer|float)" >

<!ELEMENT resources ( image_group*, review_group* ) >
<!ATTLIST resources number_image_groups %number; #REQUIRED
                    number_review_groups %number; #REQUIRED >

<!ELEMENT image_group ( image* ) >
```

```
<!ATTLIST image_group number_images %number; #REQUIRED
                      provider_name  CDATA    #REQUIRED
                      provider_image CDATA    #REQUIRED >

<!ELEMENT image ( EMPTY ) >
<!ATTLIST image url CDATA #REQUIRED >

<!ELEMENT review_group ( review* ) >
<!ATTLIST review_group number_reviews %number; #REQUIRED
                       provider_name  CDATA    #REQUIRED
                       provider_image CDATA    #REQUIRED >

<!ELEMENT review ( #PCDATA ) >
<!ATTLIST review rating   %number; #REQUIRED
                 date     CDATA #REQUIRED
                 reviewer CDATA #REQUIRED >
```

The reply to a *poi_detail_request* with the information about the POI(s) requested.

Description of elements:

Name	Type	Description
detail_field		A list of pairs (*info_field* elements) giving additional information. One pair consists of a *fieldName* and a *fieldValue*. The type of field is in the *info_type* attribute.
resources		Contains images and reviews. The images and reviwes are grouped by provider.
image_group		Contains images from one provider.
review_group		Contains reviews from one provider.
image		The url to an image.
review		Contains a review including rating.

Description of attributes:

Name	Type	Description
number_image_groups	integer	The number of image groups.
number_review_groups	integer	The number of review groups.
provider_name	string	The name of the provider from where the images or reviews are fetched.
provider_image	string	The image of the provider from where the images or reviews are fetched.
number_images	integer	The number of images.
number_reviews	integer	The number of reviews.
url	string	The url to an image. The url can be used to fetch the image thru the MC2 server.
rating	integer	The rating given by the user.
date	string	Date of the review.
reviewer	string	The user who wrote the review.

Description of enum poi_detail_t:

Name	Type	Description
dont_show	text	Hidden field not to be visible for user.
text	text	A field containing text.
street_address	text	The street address with house number.
full_address	text	The address with street, house number, zip code and zip area.
phone_number	phone_numer	The phone number.
url	url	URL to a web page.
email	email	E-mail address.
poi_url	url	Link to providers POI information.
poi_thumb	url	Thumb image information.
average_rating	integer	Average rating.
description	text	Description of a POI
open_hours	text	Open hours.
provider_info	text	Provider info. Format <Link name>;<URL>

Description of enum poi_detail_content_t:

Name	Type	Description
text		The field shall be handled as text.
phone_number		The field shall be handled as a phone number.
url		The field shall be handled as an url.
email_address		The field shall be handled as an email address.
integer		The field shall be handled as an integer.
float		The field shall be handled as a float.

46 Document Type Definition

The formal definition of the XML documents that are sent to and from the $\mathcal{MC}^2$-system.

```
<!--

MapCentral 2 (MC2) external API Document Type Definition.


Copyright (c) 1999 - 2010, Vodafone Group Services Ltd
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are p

    * Redistributions of source code must retain the above copyright notice, this list
    * Redistributions in binary form must reproduce the above copyright notice, this li
    * Neither the name of the Vodafone Group Services Ltd nor the names of its contribu

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EX



isab-mc2 is an XML language. Typical usage:

   <?xml version="1.0"?>
   <!DOCTYPE isab-mc2>

   <isab-mc2>

   ...

   </isab-mc2>


-->
```

```
<!ELEMENT isab-mc2 ( ( auth, ( user_request | route_request |
                               search_request | search_desc_request |
                               search_position_desc_request |
                               expand_request |
                               compact_search_request |
                               one_search_request |
                               poi_search_request |
                               send_sms_request |
                               user_login_request | user_verify_request |
                               user_logout_request | user_show_request |
                               user_cap_request |
                               user_favorites_request |
                               user_favorites_crc_request |
                               map_request |
                               poi_info_request | poi_detail_request |
                               email_request |
                               sms_format_request | sort_dist_request |
                               top_region_request |
                               phone_manufacturer_request |
                               phone_model_request | user_track_request |
                               user_track_add_request |
                               user_debit_log_request |
                               user_find_request | transactions_request |
                               transaction_days_request |
                               activate_request |
                               ext_services_request |
                               ext_search_request |
                               simple_poi_desc_request |
                               tunnel_request |
                               zoom_settings_request |
                               error_report |
                               poi_review_requests|
                               show_activationcode_request |
                               expand_top_region_request|
                               category_list_request |
                               copyright_strings_request |
                               ad_debit_request |
                               category_tree_request |
                               client_type_info_request |
                               server_list_for_client_type_request |
                               create_wayfinder_user_request |
                               update_hardware_keys_request |
                               get_stored_user_data_request |
                               set_stored_user_data_request |
                               friend_finder_request |
                               friend_finder_info_request |
                               cell_id_request |
                               local_category_tree_request |
                               server_info_request )+ ) |
                     ( ( user_reply | route_reply |
```

```
                search_reply | search_desc_reply |
                search_position_desc_reply |
                expand_reply |
                compact_search_reply |
                one_search_reply |
                poi_search_reply |
                send_sms_reply |
                user_login_reply | user_verify_reply |
                user_logout_reply | user_show_reply |
                user_cap_reply |
                user_favorites_reply |
                user_favorites_crc_reply |
                map_reply |
                poi_info_reply | poi_detail_reply |
                email_reply |
                sms_format_reply | sort_dist_reply |
                top_region_reply |
                phone_manufacturer_reply |
                phone_model_reply| user_track_reply |
                user_track_add_reply |
                user_debit_log_reply |
                user_find_reply | transactions_reply |
                transaction_days_reply |
                activate_reply |
                ext_services_reply |
                simple_poi_desc_reply |
                tunnel_reply |
                zoom_settings_reply |
                error_report_reply |
                poi_review_replies |
                show_activationcode_reply |
                expand_top_region_reply|
                category_list_reply |
                copyright_strings_reply |
                ad_debit_reply |
                category_tree_reply |
                client_type_info_reply |
                server_list_for_client_type_reply |
                create_wayfinder_user_reply |
                update_hardware_keys_reply |
                get_stored_user_data_reply |
                set_stored_user_data_reply |
                friend_finder_reply |
                friend_finder_info_reply |
                cell_id_reply |
                local_category_tree_reply |
                server_info_reply )+ |
              ( status_code, status_message, status_uri?,
                status_code_extended?, auth_token?, uin?,
                server_list?, server_auth_bob? ) ) ) >
```

```
<!-- General entities -->
<!ENTITY % number  "NMTOKEN">      <!-- a number, format [0-9]+ -->
<!ENTITY % bool    "(true|false)"> <!-- true or false -->
<!ENTITY % vdata   "CDATA">        <!-- attribute value -->
<!ENTITY % HREF    "%vdata;">      <!-- URI, URL or URN designating a
                                       hypertext node. -->
<!ENTITY % size_t  "%number;">     <!-- a number but inf is allowed -->
<!ENTITY % time_t  "CDATA">        <!-- Time since the Epoch
                                       (00:00:00 UTC, January 1, 1970),
                                       measured in seconds. -->
<!ENTITY % coordinate_t "CDATA">    <!-- A Latitude or longitude -->
<!ENTITY % hex_t "CDATA">           <!-- A hexadecimal number    -->
<!-- position_system_t WGS84(GPS) -->
<!ENTITY % position_system_t "(WGS84|MC2|WGS84Rad|WGS84Deg)">

<!ENTITY % image_display_type "(std|wap)">
<!ENTITY % route_image_format_t "(png|gif|wbmp)">
<!ENTITY % message_t "(html|wml|smil)">
<!ENTITY % route_turn_image_t "(map|pictogram|pictogram_set_1|
                               pictogram_set_2|pictogram_set_3|
                               pictogram_set_4|pictogram_set_5)">
<!ENTITY % overview_image_t "(image|applet|none)">
<!ENTITY % sort_distance_t "(radius|route)">
<!ENTITY % route_cost_t "(distance|time|time_with_disturbances)">
<!ENTITY % road_side_t "(left_side|right_side|unknown_side|undefined_side|
                         side_does_not_matter|left_side_exit|
                         right_side_exit)">
<!ENTITY % landmarklocation_t "(after|before|in|at|pass|into|arrive|
                               undefinedlocation)">
<!ENTITY % landmark_t "(builtUpArea|railway|area|poi|signPost|country|
                        countryAndBuiltUpArea|passedStreet|accident|
                        roadwork|camera|speedTrap|police|weather|
                        trafficGen)">
<!ENTITY % top_region_t "(country|state|internationalRegion|metaregion)">
<!ENTITY % language_t "(swedish|english|german|danish|italian|dutch|
                        spanish|french|welch|finnish|norwegian|portuguese|
                        czech|hungarian|polish|greek|american|albanian|
                        basque|catalan|frisian|irish|galician|
                        letzeburgesch|raetoRomance|serboCroatian|slovenian|
                        valencian|slovak|russian|turkish|arabic|
                        eng|swe|ger|dan|ita|dut|spa|fre|wel|fin|nor|por|
                        eng_usa|cze|alb|baq|cat|fry|gle|glg|ltz|roh|scr|
                        slv|hun|gre|pol|slo|rus|tur|ara|chi|est|lav|lit|
                        tha|bul|ind|may|isl|jpa|amh|hye|tgl|bel|ben|mya|
                        hrv|fas|gla|kat|guj|heb|hin|kan|kaz|khm|kor|lao|
                        mkd|mal|mar|mol|mon|pan|ron|srp|sin|som|swa|tam|
                        tel|bod|tir|tuk|ukr|urd|vie|zul|sot|bun|bos|sla|
                        bet|mat|scc|ukl|mlt|zh-hant|zh-hant-hk)">
```

```
<!ENTITY % route_vehicle_t "(passengercar|pedestrian|taxi)">
<!ENTITY % matchtype_t "(close|full|exact|
                         closefull|wildcard|allwords|
                         phonetic|editdistance)">
<!ENTITY % wordmatch_t "(beginning|anywhere|wildcardpart|beginningofword)">
<!ENTITY % sorttype_t "(no_sort|alfa_sort|confidence_sort)">
<!ENTITY % transactionBased_t "(no_transactions|transactions|
                                transaction_days)">
<!ENTITY % poi_info_t "(dont_show|text|url|wap_url|email|phone_number|
                        mobile_phone|fax_number|contact_info|short_info|
                        vis_address|vis_house_nbr|vis_zip_code|
                        vis_complete_zip|vis_zip_area|vis_full_address|
                        brandname|short_description|long_description|
                        citypart|state|neighborhood|open_hours|
                        nearest_train|start_date|end_date|start_time|
                        end_time|accommodation_type|check_in|check_out|
                        nbr_of_rooms|single_room_from|double_room_from|
                        triple_room_from|suite_from|extra_bed_from|
                        weekend_rate|nonhotel_cost|breakfast|
                        hotel_services|credit_card|special_feature|
                        conferences|average_cost|booking_advisable|
                        admission_charge|home_delivery|disabled_access|
                        takeaway_available|allowed_to_bring_alcohol|
                        type_food|decor|image_url|supplier|owner|
                        price_petrol_superplus|price_petrol_super|
                        price_petrol_normal|price_diesel|
                        price_biodiesel|free_of_charge|
                        tracking_data|post_address|post_zip_area|
                        post_zip_code|open_for_season|
                        ski_mountain_min_max_height|
                        snow_depth_valley_mountain|
                        snow_quality|lifts_open_total|ski_slopes_open_total|
                        cross_country_skiing_km|glacier_area|last_snowfall|
                        booking_url|booking_phone_number|
                        special_flag)" >
<!ENTITY % server_group_t "(backup|config|server|map|tmap)" >
<!ENTITY % poi_detail_t "(dont_show|text|street_address|full_address|phone_number|
                          url|email|poi_url|poi_thumb|average_rating|
                          description|open_hours|provider_info)" >
<!ENTITY % poi_detail_content_t "(text|phone_number|url|email_address|
                                  integer|float)" >



<!ENTITY % measurement_system_t "(meters|imperial|feet|yards)">

<!-- Common elements -->
<!ELEMENT status_code (#PCDATA)>
<!ELEMENT status_message (#PCDATA)>
<!ELEMENT status_uri EMPTY>
```

```
<!ATTLIST status_uri href %HREF; #REQUIRED >
<!ELEMENT status_code_extended (#PCDATA)>
<!ELEMENT name_node ( #PCDATA ) >
<!ATTLIST name_node language %language_t; #REQUIRED >


<!-- TopRegion -->
<!ELEMENT top_region ( top_region_id, boundingbox?, name_node? ) >
<!ATTLIST top_region top_region_type %top_region_t; #REQUIRED >
<!ELEMENT top_region_id ( #PCDATA ) >



<!-- Authenticate the isab-mc2 request -->
<!ELEMENT auth ( ( (auth_user, auth_passwd, user_service?, want_uin?) |
                   (user_session_id, user_session_key, user_service) |
                    auth_activate_request |
                   (uin, auth_token) |
                   (uin?, hardware_key+) ),
               pp* )>
<!ATTLIST auth indentingandlinebreaks %bool; "true"
               development %bool; "false"
               client_type CDATA #IMPLIED
               client_lang %language_t; #IMPLIED
               server_list_crc     CDATA #IMPLIED
               server_auth_bob_crc CDATA #IMPLIED
               client_source       CDATA #IMPLIED >
<!ELEMENT want_uin (#PCDATA) >
<!ELEMENT auth_user (#PCDATA)>
<!ELEMENT auth_passwd (#PCDATA)>
<!ENTITY % user_service_t "(ROUTE)">
<!ENTITY % user_method_t "(WAP|HTML|NAV|XML|SMS|OPERATOR)">
<!ELEMENT user_service (#PCDATA)>
<!ELEMENT auth_activate_request EMPTY>
<!ELEMENT server_list ( server_group+ ) >
<!ATTLIST server_list crc         CDATA    #REQUIRED
                      numberitems %number; #REQUIRED >
<!ELEMENT server_group (server+)>
<!ATTLIST server_group  type           %server_group_t; #REQUIRED
                        numberitems            %number; #REQUIRED
                        switch_group_threshold %number; #REQUIRED>
<!ELEMENT server (#PCDATA) >
<!ELEMENT server_auth_bob (#PCDATA) >
<!ATTLIST server_auth_bob crc  CDATA #REQUIRED >
<!ELEMENT pp ( #PCDATA ) >
<!ATTLIST pp key CDATA #REQUIRED >


<!-- User request -->
<!ELEMENT user_request (user)>
```

```
<!ATTLIST user_request transaction_id ID #REQUIRED
                       new_user %bool; #IMPLIED >


<!ELEMENT user ( user_id?, first_name?, last_name?, initials?,
                 language?, measurement_system?,
                 email_address?, operator_comment?,
                 search_for_municipal?, search_for_city?,
                 search_for_citypart?, search_for_zipcode?,
                 search_for_ziparea?,
                 search_for_street?,
                 search_for_company?, search_for_category?,
                 search_for_misc?,
                 new_password?, old_password?,
                 service*, phone*, user_licence_key*,
                 binary_key*, region_access*,
                 wayfinder_subscription?, right*, token*, pin*,
                 id_key*, last_client* ) >
<!ATTLIST user uin CDATA #IMPLIED
               birth_date CDATA #IMPLIED
               route_cost %route_cost_t; #IMPLIED
               route_vehicle %route_vehicle_t; #IMPLIED
               search_match_type %matchtype_t; #IMPLIED
               search_word_match_type %wordmatch_t; #IMPLIED
               search_sort_type %sorttype_t; #IMPLIED
               valid_date %time_t; #IMPLIED
               edit_user_right %bool; #IMPLIED
               address1 CDATA #IMPLIED
               address2 CDATA #IMPLIED
               address3 CDATA #IMPLIED
               address4 CDATA #IMPLIED
               address5 CDATA #IMPLIED
               route_turn_image %route_turn_image_t; #IMPLIED
               overview_image_type %overview_image_t; #IMPLIED
               transactionBased %transactionBased_t; #IMPLIED
               deviceChanges %number; #IMPLIED
               supportComment CDATA #IMPLIED
               postalCity CDATA #IMPLIED
               zipCode CDATA #IMPLIED
               companyName CDATA #IMPLIED
               companyReference CDATA #IMPLIED
               companyVATNbr CDATA #IMPLIED
               emailBounces %number; #IMPLIED
               addressBounces %number; #IMPLIED
               customerContactInfo CDATA #IMPLIED
               brand_origin CDATA #IMPLIED
               brand CDATA #IMPLIED >

<!ELEMENT user_id (#PCDATA)>
<!ELEMENT first_name (#PCDATA)>
```

```
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT initials (#PCDATA)>
<!ELEMENT default_transportation (#PCDATA)>
<!ELEMENT language (#PCDATA)>
<!ELEMENT measurement_system (#PCDATA)>
<!ELEMENT email_address (#PCDATA)>
<!ELEMENT operator_comment ( #PCDATA ) >
<!ELEMENT new_password (#PCDATA)>
<!ELEMENT old_password (#PCDATA)>

<!ELEMENT service (service_type, service_method, service_delete?)>
<!ELEMENT service_type (#PCDATA)>
<!ELEMENT service_method (#PCDATA)>
<!ELEMENT service_delete EMPTY>

<!ELEMENT phone (phone_number, phone_manufacturer?, phone_model?,
                 phone_delete?)>
<!ELEMENT phone_number (#PCDATA)>
<!ELEMENT phone_manufacturer (#PCDATA)>
<!ELEMENT phone_model (#PCDATA)>
<!ELEMENT phone_delete EMPTY>

<!ELEMENT binary_key ( key_data, key_delete? ) >
<!ATTLIST binary_key id CDATA #REQUIRED >
<!ELEMENT key_data ( #PCDATA ) >
<!ELEMENT key_delete EMPTY >

<!ENTITY % hardware_key_type_t "(imei|btmac|bbpin|imsi|esn|
                                 phone_msisdn|iphone_dev_id|
                                 customer_msisdn)">
<!ELEMENT user_licence_key EMPTY >
<!ATTLIST user_licence_key id       CDATA                 #REQUIRED
                           key      CDATA                 #REQUIRED
                           key_type %hardware_key_type_t; #IMPLIED
                           product  CDATA                 #IMPLIED
                           delete   %bool;                #IMPLIED >

<!ELEMENT region_access ( region_access_delete? ) >
<!ATTLIST region_access id           %number; #REQUIRED
                        top_region_id %number; #REQUIRED
                        start_time    %time_t; #REQUIRED
                        end_time      %time_t; #REQUIRED >
<!ELEMENT region_access_delete EMPTY>

<!ELEMENT wayfinder_subscription ( wayfinder_subscription_delete? )>
<!ATTLIST wayfinder_subscription id      %number; #REQUIRED
                                 type    %number; #REQUIRED >
<!ELEMENT wayfinder_subscription_delete EMPTY>

<!ELEMENT right EMPTY >
```

```
<!ATTLIST right         id            %number; #REQUIRED
                        add_time      %time_t; #IMPLIED
                        type          %number; #IMPLIED
                        top_region_id %number; #IMPLIED
                        start_time    %time_t; #IMPLIED
                        end_time      %time_t; #IMPLIED
                        deleted       %bool;   #IMPLIED
                        origin        CDATA    #IMPLIED >

<!ELEMENT token ( delete? ) >
<!ATTLIST token id            %number; #REQUIRED
                create_time   %time_t; #IMPLIED
                age           %number; #IMPLIED
                token         CDATA    #IMPLIED
                group         CDATA    #IMPLIED >

<!ELEMENT pin ( delete? ) >
<!ATTLIST pin id            %number; #REQUIRED
              PIN           CDATA    #IMPLIED
              comment       CDATA    #IMPLIED >
<!ELEMENT delete EMPTY >

<!ELEMENT id_key ( delete? ) >
<!ENTITY % id_key_t "(account|hardware|hardware_and_time|
                      service_id_and_time|client_type_and_time)">
<!ATTLIST id_key id            %number;      #REQUIRED
                 type          %id_key_t;    #IMPLIED
                 key           CDATA         #IMPLIED >

<!ELEMENT last_client EMPTY >
<!ATTLIST last_client id                  %number; #REQUIRED
                      client_type         CDATA    #IMPLIED
                      client_type_options CDATA    #IMPLIED
                      version             CDATA    #IMPLIED
                      extra               CDATA    #IMPLIED
                      origin              CDATA    #IMPLIED
                      history             %bool;   #IMPLIED
                      changer_uin         CDATA    #IMPLIED
                      change_time         %time_t; #IMPLIED >


<!ELEMENT favorite ( position_item, fav_info* )>
<!ATTLIST favorite
                    id               CDATA  #REQUIRED
                    name             CDATA  #REQUIRED
                    short_name       CDATA  #REQUIRED
                    description      CDATA  #REQUIRED
                    category         CDATA  #REQUIRED
                    map_icon_name    CDATA  #REQUIRED >
```

```
<!ELEMENT fav_info EMPTY >
<!ATTLIST fav_info  type  %poi_info_t; #REQUIRED
                    key   CDATA        #REQUIRED
                    value CDATA        #REQUIRED >

<!-- User reply -->
<!ELEMENT user_reply (status_code, status_message, status_code_extended?)>
<!ATTLIST user_reply transaction_id ID #REQUIRED>


<!-- User show request -->
<!ELEMENT user_show_request ( (user_id |
                               (user_session_id, user_session_key))? ) >
<!ENTITY % user_show_t "(all|active)">
<!ATTLIST user_show_request transaction_id ID #REQUIRED
                            uin CDATA #IMPLIED
                            cached_data %bool; #IMPLIED
                            show        %user_show_t; "all" >


<!-- User show reply -->
<!ELEMENT user_show_reply ( ( user ) |
                            ( status_code, status_message,
                              status_code_extended? ) )>
<!ATTLIST user_show_reply transaction_id ID #REQUIRED>

<!--- User cap request -->
<!ELEMENT user_cap_request EMPTY >
<!ATTLIST user_cap_request transaction_id ID #REQUIRED >

<!--- User cap reply -->
<!ELEMENT user_cap_reply (user_id, cap*, pin*, popup*) >
<!ATTLIST user_cap_reply transaction_id ID #REQUIRED >
<!ENTITY % cap_name_type "(gps|locator|route|fleet|traffic)">
<!ELEMENT cap EMPTY >
<!ATTLIST cap name %cap_name_type; #REQUIRED >
<!ELEMENT popup (popup_message, popup_once?, popup_url?)>
<!-- Yes No if url and attr for if to exit if no on url. -->
<!ENTITY % popup_url_t "(yes_no|goto_or_exit)">
<!ELEMENT popup_message ( #PCDATA )>
<!ELEMENT popup_once ( #PCDATA )>
<!ELEMENT popup_url ( #PCDATA )>
<!ATTLIST popup_url url_type %popup_url_t; "yes_no" >


<!-- User favorites request -->
<!ELEMENT user_favorites_request ( (user_id | uin |
                                    (user_session_id, user_session_key) )?,
                                   favorite_id_list?,
                                   delete_favorite_id_list?,
```

```
                                  add_favorite_list?,
                                  auto_dest_favorite? )>
<!ATTLIST user_favorites_request
                              transaction_id ID #REQUIRED
                              fetch_auto_dest   %bool; "false"
                              sync_favorites    %bool; "true"
                              position_system   %position_system_t; "MC2"
                              fav_info_in_desc  %bool; "true" >
<!ELEMENT favorite_id_list ( favorite_id* )>
<!ELEMENT favorite_id ( #PCDATA )>
<!ELEMENT delete_favorite_id_list ( favorite_id+ )>
<!ELEMENT add_favorite_list ( favorite+ )>
<!ELEMENT auto_dest_favorite ( favorite? )>


<!-- User favorites reply -->
<!ELEMENT user_favorites_reply ( (delete_favorite_id_list?,
                                  add_favorite_list?,
                                  auto_dest_favorite?) |
                                 ( status_code, status_message,
                                   status_code_extended? ) )>
<!ATTLIST user_favorites_reply transaction_id ID #REQUIRED
                               crc CDATA #REQUIRED >

<!-- user_favorites_crc_request -->
<!ELEMENT user_favorites_crc_request EMPTY >
<!ATTLIST user_favorites_crc_request transaction_id ID #REQUIRED
                                     crc CDATA #REQUIRED >

<!-- user_favorites_crc_reply -->
<!ELEMENT user_favorites_crc_reply EMPTY >
<!ATTLIST user_favorites_crc_reply transaction_id ID #REQUIRED
                                   crc_match %bool; #REQUIRED >

<!-- Route Request -->
<!ELEMENT route_request ( route_request_header,
                          routeable_item_list,
                          routeable_item_list )>
<!ATTLIST route_request transaction_id ID #REQUIRED>

<!ELEMENT route_request_header ( route_preferences )>
<!ENTITY % reroute_reason_t "(unknown|truncated_route|off_track|
                              traffic_info_update|user_request)" >
<!ATTLIST route_request_header
          previous_route_id CDATA  #IMPLIED
          reroute_reason    %reroute_reason_t;  #IMPLIED>


<!ELEMENT route_preferences ( ( user_id  | route_settings | uin |
                                (user_session_id, user_session_key) ),
```

```
                              image_settings? )>
<!ENTITY % route_description_type_t "(normal|compact)">
<!ATTLIST route_preferences
          route_description_type %route_description_type_t; #REQUIRED
          route_image_links %bool; "false"
          route_overview_image_width %number; "256"
          route_overview_image_height %number; "256"
          route_turn_image_width %number; "256"
          route_turn_image_height %number; "256"
          route_image_default_format %route_image_format_t; "png"
          route_image_display_type %image_display_type; "std"
          route_turn_data %bool; "false"
          route_boundingbox_position_sytem %position_system_t; "MC2"
          route_turn_boundingbox %bool; "false"
          route_road_data %bool; "false"
          route_items %bool; "true"
          abbreviate_route_names %bool; "true"
          route_landmarks %bool; "false"
          route_measurment_system %measurement_system_t; "meters">
<!ELEMENT route_settings ( route_costA?,
                           route_costB?,
                           route_costC?,
                           language )>
<!ATTLIST route_settings route_vehicle %route_vehicle_t; #REQUIRED
                         avoid_toll_road %bool; #IMPLIED
                         avoid_highway   %bool; #IMPLIED >
<!ELEMENT route_costA ( #PCDATA )>
<!ELEMENT route_costB ( #PCDATA )>
<!ELEMENT route_costC ( #PCDATA )>
<!ELEMENT routeable_item_list ( (position_item | search_item)+ )>


<!-- Route Reply -->
<!ELEMENT route_reply ( ( route_reply_header,
                          route_origin, route_destination,
                          route_reply_items ) |
                        ( status_code, status_message, status_uri?,
                          status_code_extended? ) )>
<!ATTLIST route_reply transaction_id ID        #REQUIRED
                      route_id       CDATA     #REQUIRED
                      ptui           %number;  #IMPLIED >

<!ELEMENT boundingbox EMPTY>
<!ATTLIST boundingbox
             position_sytem %position_system_t; #REQUIRED
             north_lat CDATA #REQUIRED
             west_lon  CDATA #REQUIRED
             south_lat CDATA #REQUIRED
             east_lon  CDATA #REQUIRED >
```

```
<!ELEMENT route_reply_header ( total_distance,
                               total_distance_nbr,
                               total_time,
                               total_time_nbr,
                               total_standstilltime,
                               total_standstilltime_nbr,
                               average_speed,
                               average_speed_nbr,
                               routing_vehicle,
                               routing_vehicle_type,
                               boundingbox,
                               route_overview_link?,
                               route_overview_width?,
                               route_overview_height? )>
<!ELEMENT total_distance ( #PCDATA )>
<!ELEMENT total_distance_nbr ( #PCDATA )>
<!ELEMENT total_time ( #PCDATA )>
<!ELEMENT total_time_nbr ( #PCDATA )>
<!ELEMENT total_standstilltime ( #PCDATA )>
<!ELEMENT total_standstilltime_nbr ( #PCDATA )>
<!ELEMENT average_speed ( #PCDATA )>
<!ELEMENT average_speed_nbr ( #PCDATA )>
<!ELEMENT routing_vehicle ( #PCDATA )>
<!ELEMENT routing_vehicle_type ( #PCDATA )>
<!ELEMENT route_overview_link ( #PCDATA )>
<!ELEMENT route_overview_width ( #PCDATA )>
<!ELEMENT route_overview_height ( #PCDATA )>
<!ELEMENT route_origin ( search_item+ )>
<!ELEMENT route_destination ( search_item+ )>
<!ELEMENT route_reply_items ( route_reply_item* )>
<!ELEMENT route_reply_item ( description?,
                             turn?,
                             distance?,
                             time?,
                             roadname?,
                             exitcount?,
                             signposttext?,
                             signpostexitnbr?,
                             signpostroutenbr?,
                             start_dir?,
                             route_housenumber_start_direction?,
                             transporation_type?,
                             crossing_type?,
                             route_turn_link?,
                             route_turn_width?,
                             route_turn_height?,
                             boundingbox?,
                             position_item?,
                             route_road_item*,
                             route_landmark_item* )>
```

```
<!ATTLIST route_reply_item   controlled_access   %bool; #IMPLIED
                             ramp               %bool; #IMPLIED
                             roundabout         %bool; #IMPLIED
                             drive_on_right_side %bool; #IMPLIED >
<!ELEMENT description ( #PCDATA )>
<!ENTITY % route_turn_t "(left|right|ahead|u_turn|followroad|
                          enter_roundabout|exit_roundabout|
                          ahead_roundabout|right_roundabout|
                          left_roundabout|off_ramp|on_ramp|
                          enter_bus|exit_bus|change_bus|
                          park_car|start|finally|exit|
                          keep_left|keep_right|
                          enter_ferry|exit_ferry|change_ferry|
                          start_with_u_turn|u_turn_roundabout|
                          endofroad_left_turn|endofroad_right_turn|
                          off_ramp_left|off_ramp_right|
                          on_main|off_main|
                          no_turn)">
<!ENTITY % crossing_t "undefined_crossing|no_crossing|
                       crossing_3ways_t|crossing_3ways_y|crossing_4ways|
                       crossing_5ways|crossing_6ways|crossing_7ways|
                       crossing_8ways|crossing_2roundabout|
                       crossing_3roundabout|crossing_4roundabout|
                       crossing_4roundabout_asymmetric|
                       crossing_5roundabout|crossing_6roundabout|
                       crossing_7roundabout" >
<!ENTITY % route_start_dir_t "(north|northnortheast|northeast|
                               eastnortheast|east|eastsoutheast|
                               southeast|southsoutheast|south|
                               southsouthwest|southwestwestsouthwest|
                               west|westnorthwest|northwest|
                               northnorthwest)" >
<!ENTITY % route_housenumber_start_direction_t "(leftodd|rightodd|
                                                 increasing|decreasing|
                                                 unknown)">
<!ENTITY % route_transportation_t "(drive|walk|bus)">
<!ELEMENT turn ( #PCDATA )>
<!ELEMENT distance ( #PCDATA )>
<!ELEMENT time ( #PCDATA )>
<!ELEMENT roadname ( #PCDATA )>
<!ELEMENT exitcount ( #PCDATA )>
<!ELEMENT signposttext ( #PCDATA )>
<!ELEMENT signpostexitnbr ( #PCDATA )>
<!ELEMENT signpostroutenbr ( #PCDATA )>
<!ELEMENT start_dir ( #PCDATA )>
<!ELEMENT route_housenumber_start_direction ( #PCDATA )>
<!ELEMENT transporation_type ( #PCDATA )>
<!ELEMENT crossing_type ( #PCDATA )>
<!ELEMENT route_turn_link ( #PCDATA )>
<!ELEMENT route_turn_width ( #PCDATA )>
```

```
<!ELEMENT route_turn_height ( #PCDATA )>
<!ELEMENT route_road_item ( (lat, lon)+ ) >
<!ATTLIST route_road_item
                            speedLimit %number; #REQUIRED
                            is_turn %bool;      #IMPLIED
                            controlled_access   %bool; #IMPLIED
                            ramp                %bool; #IMPLIED
                            roundabout          %bool; #IMPLIED
                            drive_on_right_side %bool; #IMPLIED >
<!ELEMENT route_landmark_item ( description, road_side?,
                                landmarklocation_type?, landmark_type?,
                                distance?, name? )>
<!ATTLIST route_landmark_item at_turn   %bool; #REQUIRED
                              is_detour %bool; #IMPLIED
                              is_start  %bool; #IMPLIED
                              is_stop   %bool; #IMPLIED >
<!ELEMENT road_side ( #PCDATA )>
<!ELEMENT landmarklocation_type ( #PCDATA )>
<!ELEMENT landmark_type ( #PCDATA )>


<!-- Search Request -->
<!ELEMENT search_request ( search_request_header,
                           (search_query | proximity_query ) )>
<!ATTLIST search_request transaction_id ID #REQUIRED>

<!ELEMENT search_request_header (search_preferences,
                                 search_explicit_itemid?)>
<!ATTLIST search_request_header
                         position_sytem %position_system_t; "MC2"
                         position_search_items %bool; "false"
                         position_search_areas %bool; "false"
                         search_area_starting_index %number; "0"
                         search_area_ending_index %number; "49"
                         search_item_starting_index %number; "0"
                         search_item_ending_index %number; "99"
                         full_search_area_match_purge %bool; #IMPLIED >

<!ELEMENT search_preferences ( ((user_id|uin), search_settings?) |
                               search_settings |
                               (user_session_id, user_session_key,
                                search_settings?) )>
<!ELEMENT search_explicit_itemid EMPTY>
<!ELEMENT search_settings ( search_for_municipal?, search_for_city?,
                            search_for_citypart?, search_for_zipcode?,
                            search_for_ziparea?,
                            search_for_street?,
                            search_for_company?, search_for_category?,
                            search_for_misc?,
                            show_search_area_municipal?,
```

```
                            show_search_area_city?,
                            show_search_area_city_part?,
                            show_search_area_zipcode?,
                            show_search_area_ziparea?,
                            show_search_area_country?,
                            show_search_item_municipal?,
                            show_search_item_city?,
                            show_search_item_city_part?,
                            show_search_item_zipcode?,
                            show_search_item_ziparea?,
                            language? )>
<!ATTLIST search_settings
              matchtype                        %matchtype_t; #IMPLIED
              wordmatch                        %wordmatch_t; #IMPLIED
              sorttype                         %sorttype_t;  #IMPLIED
              minimum_numberhits               %number;      #IMPLIED >
<!ELEMENT search_for_municipal EMPTY>
<!ELEMENT search_for_city EMPTY>
<!ELEMENT search_for_citypart EMPTY>
<!ELEMENT search_for_zipcode EMPTY>
<!ELEMENT search_for_ziparea EMPTY>
<!ELEMENT search_for_street EMPTY>
<!ELEMENT search_for_company EMPTY>
<!ELEMENT search_for_category EMPTY>
<!ELEMENT search_for_misc EMPTY>
<!ELEMENT show_search_area_municipal EMPTY>
<!ELEMENT show_search_area_city EMPTY>
<!ELEMENT show_search_area_city_part EMPTY>
<!ELEMENT show_search_area_zipcode EMPTY>
<!ELEMENT show_search_area_ziparea EMPTY>
<!ELEMENT show_search_area_country EMPTY>
<!ELEMENT show_search_item_municipal EMPTY>
<!ELEMENT show_search_item_city EMPTY>
<!ELEMENT show_search_item_city_part EMPTY>
<!ELEMENT show_search_item_zipcode EMPTY>
<!ELEMENT show_search_item_ziparea EMPTY>

<!ELEMENT search_query ( top_region?, (search_area_query | search_area),
                         search_item_query? )>
<!ELEMENT search_area_query ( #PCDATA )>
<!ELEMENT search_item_query ( #PCDATA )>
<!ATTLIST search_item_query house_number CDATA #IMPLIED >
<!ELEMENT proximity_query ( ( ((search_item | position_item), distance?) |
                              boundingbox),
                            search_item_query? ) >


<!-- Search Reply -->
<!ELEMENT search_reply ( ( search_area_list?, search_item_list? ) |
                         ( status_code, status_message,
```

```
                            status_code_extended? ) )>
<!ATTLIST search_reply transaction_id ID #REQUIRED>

<!ELEMENT search_area_list ( search_area* )>
<!ATTLIST search_area_list numberitems %number; #REQUIRED
                           total_numberitems %number; #IMPLIED
                           starting_index %number; #IMPLIED
                           ending_index %number; #IMPLIED >

<!ELEMENT search_item_list ( search_item* )>
<!ATTLIST search_item_list numberitems %number; #REQUIRED
                           total_numberitems %number; #IMPLIED
                           starting_index %number; #IMPLIED
                           ending_index %number; #IMPLIED >

<!-- General search elements -->
<!ELEMENT name ( #PCDATA )>

<!-- Search Item -->

<!ENTITY % search_item_type_t "(street|pointofinterest|category|misc|person|
                                other)">
<!ELEMENT search_item ( name, itemid, streetnbr?, explicit_itemid?,
                        location_name?, lat?, lon?, category_list?,
                        boundingbox?, search_area*, info_item? )>
<!ATTLIST search_item search_item_type %search_item_type_t; #REQUIRED
                      image CDATA #IMPLIED
                      advert %bool; #IMPLIED
                      top_region_id %number; #IMPLIED >
<!ELEMENT itemid ( #PCDATA )>
<!ELEMENT streetnbr ( #PCDATA )>
<!ELEMENT explicit_itemid ( #PCDATA )>
<!ELEMENT location_name ( #PCDATA )>

<!-- Search Area -->

<!ENTITY % search_area_type_t "(municipal|city|citypart|zipcode|ziparea|
                                country|other)">
<!ELEMENT search_area ( name, areaid, location_name?,
                        lat?, lon?, boundingbox?, top_region_id?,
                        search_area* )>
<!ATTLIST search_area search_area_type %search_area_type_t; #REQUIRED>
<!ELEMENT areaid ( #PCDATA )>

<!-- Position Item -->
<!ELEMENT position_item ( lat, lon, angle? )>
<!ATTLIST position_item position_system %position_system_t; #REQUIRED>
<!-- latitude and longitude WSG84 format: (N|S|E|W) D(D*)MMSS[.ddd] -->
<!ELEMENT lat ( #PCDATA )>
<!ELEMENT lon ( #PCDATA )>
```

```
<!ELEMENT angle ( #PCDATA ) >

<!ELEMENT image_name ( #PCDATA ) >
<!ELEMENT type ( #PCDATA ) >

<!ELEMENT search_position_desc_request ( position_item ) >
<!ATTLIST search_position_desc_request
          transaction_id ID #REQUIRED
          language CDATA #REQUIRED >

<!ELEMENT search_position_desc_reply ( top_region?, search_hit_type* ) >
<!ATTLIST search_position_desc_reply transaction_id ID #REQUIRED
                                    length %number; #REQUIRED >

<!ELEMENT search_desc_request EMPTY >
<!ATTLIST search_desc_request transaction_id ID #REQUIRED
                              crc %hex_t; #REQUIRED
                              language CDATA #REQUIRED
                              uin %number; #IMPLIED
                              desc_version %number; "0" >

<!ELEMENT search_desc_reply (search_hit_type* | crc_ok) >
<!ATTLIST search_desc_reply transaction_id ID #REQUIRED
                            crc %hex_t; #REQUIRED
                            length %number; #REQUIRED >

<!ELEMENT search_hit_type (name, top_region_id?, image_name?, type?) >
<!ATTLIST search_hit_type round %number; #REQUIRED
                          heading %number; #REQUIRED >

<!ELEMENT category_query ( #PCDATA ) >
<!ELEMENT category_id ( #PCDATA ) >
<!ELEMENT category_list (category_id+) >

<!ELEMENT compact_search_request ( search_item_query,
                                   ( category_query | category_name  |
                                     category_id | category_list )?,
                                   ( (search_area_query, top_region_id)|
                                     search_area|
                                    (position_item, distance?) ) ) >
<!ATTLIST compact_search_request transaction_id ID #REQUIRED
                          start_index %number; #REQUIRED
                          end_index %number; #REQUIRED
                          max_hits %number; #REQUIRED
                          language CDATA #REQUIRED
                          round %number; #IMPLIED
                          heading %number; #IMPLIED
                          uin %number; #IMPLIED
                          version %number; "0"
                          category_search %bool; "false"
```

```
                        include_category_id %bool; "false"
                        include_top_region_id %bool; "false"
                        use_persistent_ids %bool; "false"
                        include_info_item %bool; "false"
                        position_system %position_system_t; "MC2" >

<!ELEMENT ad_results_text ( #PCDATA ) >
<!ELEMENT all_results_text ( #PCDATA ) >

<!ELEMENT compact_search_reply (
                ( ad_results_text?, all_results_text?, search_hit_list* ) |
                ( status_code, status_message,
                  status_uri?, status_code_extended? ) )>
<!ATTLIST compact_search_reply transaction_id ID #REQUIRED>

<!-- Category Handling -->
<!ELEMENT category_list_request ( position_item? ) >
<!ATTLIST category_list_request transaction_id ID #REQUIRED
                                crc %hex_t; #REQUIRED
                                language CDATA #REQUIRED >

<!ELEMENT category_list_reply (cat*|crc_ok) >
<!ATTLIST category_list_reply transaction_id ID #REQUIRED
                              count %number; #REQUIRED
                              crc %hex_t; #REQUIRED >

<!ELEMENT poi_search_request (position_item, distance,
                              category_list?,
                              search_item_query?) >

<!ATTLIST poi_search_request transaction_id ID #REQUIRED
                             start_index %number; #REQUIRED
                             end_index %number; #REQUIRED
                             language CDATA #REQUIRED
                             include_top_region_id %bool; "false"
                             use_persistent_ids %bool; "false"
                             include_info_item %bool; "false" >

<!ATTLIST poi_search_reply transaction_id ID #REQUIRED >
<!ELEMENT poi_search_reply ( search_item_list |
          ( status_code, status_message,
          status_uri?, status_code_extended? ) )>

<!-- Note: change name of category node -->
<!ELEMENT category (name,category_name,image_name) >

<!ELEMENT cat (name,image_name?,cat*) >
<!ATTLIST cat cat_id %number; #IMPLIED >

<!ELEMENT category_tree ( cat* ) >
```

```
<!ENTITY % category_tree_t "(vicinity|eventfinder)" >
<!ATTLIST category_tree_request transaction_id ID #REQUIRED
                                crc %hex_t; #REQUIRED
                                language CDATA #REQUIRED
                                type %category_tree_t; "vicinity" >

<!ELEMENT category_tree_request EMPTY>

<!ELEMENT category_tree_reply (category_tree|crc_ok |
                               ( status_code, status_message,
                                status_code_extended? ) ) >
<!ATTLIST category_tree_reply transaction_id ID #REQUIRED
                              crc %hex_t; #REQUIRED >


<!-- Copyright Strings Request -->

<!ELEMENT copyright_strings_request EMPTY >
<!ATTLIST copyright_strings_request transaction_id ID #REQUIRED
                                    crc %hex_t; #REQUIRED
                                    language CDATA #REQUIRED >
<!ELEMENT copyright_strings_data ( #PCDATA )>

<!ELEMENT copyright_strings_reply ( crc_ok | copyright_strings_data ) >
<!ATTLIST copyright_strings_reply transaction_id ID #REQUIRED
                                  crc %hex_t; #REQUIRED >



<!ELEMENT search_hit_list ( ad_results_text?, all_results_text?, (search_item* | searcl
<!ATTLIST search_hit_list numberitems %number; #REQUIRED
                          total_numberitems %number; #REQUIRED
                          starting_index %number; #REQUIRED
                          ending_index %number; #REQUIRED
                          heading %number; #REQUIRED
                          top_hits %number; "0" >

<!-- Expand Request -->
<!-- When expanding categories search_area must be provided.
     search_area is not needed when expanding a pointofinterest.-->
<!ELEMENT expand_request ( expand_request_header,
                           expand_request_query )>
<!ATTLIST expand_request transaction_id ID #REQUIRED>
<!ELEMENT expand_request_header (search_preferences)>
<!ENTITY % expand_request_location_t "(all_possible|country_city)">
<!ATTLIST expand_request_header
                         position_system %position_system_t; "MC2"
                         include_top_region_id %bool; "false"
                         location_name %expand_request_location_t; #IMPLIED >
```

```
<!ELEMENT expand_request_query ( (search_area, search_item) |
                                 search_item | position_item | search_area )>


<!-- Expand Reply -->
<!ELEMENT expand_reply ( (search_item_list | search_area_list |
                          companydata)+ |
                         (status_code, status_message,
                          status_code_extended? ) )>
<!ATTLIST expand_reply transaction_id ID #REQUIRED>
<!-- More later -->
<!ELEMENT companydata EMPTY>

<!-- ad debit request -->
<!ELEMENT ad_debit (itemid) >
<!ATTLIST ad_debit type %number; #REQUIRED >

<!ELEMENT ad_debit_request (ad_debit*) >
<!ATTLIST ad_debit_request transaction_id ID #REQUIRED
                           count %number; #REQUIRED >


<!-- ad debit reply -->
<!ELEMENT ad_debit_reply EMPTY >
<!ATTLIST ad_debit_reply transaction_id ID #REQUIRED >

<!-- Send SMS Request -->
<!ELEMENT send_sms_request ( phone_number,
                             (smsmessage |
                              (route_sms_message, route_message_data) |
                              (local_map_sms_settings, local_map_data) |
                              (wayfinder_destination_sms) |
                              (wayfinder_route_sms) |
                              (wayfinder_favourite_sms) |
                              (wap_push_service_indication) ) ) >
<!ATTLIST send_sms_request transaction_id ID #REQUIRED
                           wayfinder_sms_version CDATA "1">
<!ELEMENT smsmessage (#PCDATA)> <!-- Max 160 characters -->
<!ELEMENT route_sms_message ( phone_manufacturer, phone_model ) >
<!ATTLIST route_sms_message wap_link %bool; #REQUIRED >
<!-- Route message data -->
<!ELEMENT route_message_data ( language, signature,
                               originString, originLocationString,
                               destinationString,
                               destinationLocationString ) >
<!ATTLIST route_message_data route_id CDATA  #REQUIRED >
<!ELEMENT signature ( #PCDATA ) >
<!ELEMENT originString ( #PCDATA ) >
<!ELEMENT originLocationString ( #PCDATA ) >
<!ELEMENT destinationString ( #PCDATA ) >
```

```
<!ELEMENT destinationLocationString ( #PCDATA ) >
<!-- Local map settings -->
<!ELEMENT local_map_sms_settings ( phone_manufacturer, phone_model ) >
<!-- Wayfinder destination sms -->
<!ELEMENT wayfinder_destination_sms ( position_item, signature? ) >
<!ATTLIST wayfinder_destination_sms description CDATA #REQUIRED >
<!-- Wayfinder route sms -->
<!ELEMENT wayfinder_route_sms ( position_item, position_item, signature? )>
<!ATTLIST wayfinder_route_sms orig_description CDATA #REQUIRED
                              dest_description CDATA #REQUIRED >
<!-- Wayfinder favourite sms -->
<!ELEMENT wayfinder_favourite_sms ( position_item, name, short_name,
                                    category_name, map_icon_name,
                                    signature? )>
<!ATTLIST wayfinder_favourite_sms description CDATA #REQUIRED >
<!ELEMENT short_name ( #PCDATA ) >
<!ELEMENT category_name ( #PCDATA ) >
<!ELEMENT map_icon_name ( #PCDATA ) >

<!-- WAP Push, Service Indication (WAP-167-ServiceInd-20010731-a) -->
<!ELEMENT wap_push_service_indication ( #PCDATA ) >
<!ATTLIST wap_push_service_indication href %HREF; #REQUIRED >

<!-- Send SMS Reply -->
<!ELEMENT send_sms_reply ( status_code, status_message,
                           status_code_extended? )>
<!ATTLIST send_sms_reply transaction_id ID #REQUIRED>


<!-- User login Request -->
<!ELEMENT user_login_request ( user_name, user_password,
                               user_service? ) >
<!ATTLIST user_login_request
            transaction_id ID          #REQUIRED
            user_create_session %bool; "false"
            client_type         CDATA  #IMPLIED
            client_type_options CDATA  #IMPLIED >
<!ELEMENT user_name ( #PCDATA ) >
<!ELEMENT user_password ( #PCDATA ) >

<!-- User login Reply -->
<!ELEMENT user_login_reply ( status_code, status_message,
                             status_code_extended?,
                             user_session_id?, user_session_key? ) >
<!ATTLIST user_login_reply transaction_id ID #REQUIRED>
<!ELEMENT user_session_id ( #PCDATA ) >
<!ELEMENT user_session_key ( #PCDATA ) >


<!-- User verify Request -->
```

```
<!ELEMENT user_verify_request ( user_session_id, user_session_key ) >
<!ATTLIST user_verify_request transaction_id ID #REQUIRED>

<!-- User verify Reply -->
<!ELEMENT user_verify_reply ( status_code, status_message,
                              status_code_extended? ) >
<!ATTLIST user_verify_reply transaction_id ID #REQUIRED>


<!-- User logout Request -->
<!ELEMENT user_logout_request ( user_session_id, user_session_key ) >
<!ATTLIST user_logout_request transaction_id ID #REQUIRED>

<!-- User logout Reply -->
<!ELEMENT user_logout_reply ( status_code, status_message,
                              status_code_extended? ) >
<!ATTLIST user_logout_reply transaction_id ID #REQUIRED>


<!-- Map request -->
<!ELEMENT map_request ( map_request_header, map_symbol_list? ) >
<!ATTLIST map_request transaction_id ID #REQUIRED>
<!ELEMENT map_request_header ( boundingbox, image_settings?,
                               route_data?, phone_position? ) >
<!ATTLIST map_request_header
             image_width  %number; "400"
             image_height %number; "400"
             image_default_format %route_image_format_t; "png"
             image_display_type %image_display_type; "std"
             showMap             %bool;   "true"
             showTopographMap    %bool;   "true"
             showPOI             %bool;   "true"
             showRoute           %bool;   "true"
             showScale           %bool;   "false"
             showTraffic         %bool;   "false" >
<!ELEMENT image_settings EMPTY>
<!ATTLIST image_settings
             image_show_street_main %bool; "true"
             image_show_street_first %bool; "true"
             image_show_street_second %bool; "true"
             image_show_street_third %bool; "true"
             image_show_street_fourth %bool; "true"
             image_show_builtup_area %bool; "true"
             image_show_park %bool; "true"
             image_show_forest %bool; "true"
             image_show_building %bool; "true"
             image_show_water %bool; "true"
             image_show_island %bool; "true"
             image_show_pedestrianarea %bool; "true"
             image_show_aircraftroad %bool; "true"
```

```
            image_show_land %bool; "true" >
<!ELEMENT route_data ( route_id, route_turn? ) >
<!ELEMENT route_id ( #PCDATA ) >
<!ELEMENT route_turn ( #PCDATA ) >
<!ELEMENT map_symbol_list ( map_symbol_item+ ) >
<!ELEMENT map_symbol_item ( position_item, name ) >
<!ATTLIST map_symbol_item href %HREF; #REQUIRED>

<!ELEMENT phone_position ( lat, lon,
                           innerRadius, outerRadius,
                           startAngle, stopAngle,
                           levelOfConfidence ) >
<!ATTLIST phone_position position_sytem %position_system_t; "MC2" >
<!ELEMENT innerRadius ( #PCDATA )>
<!ELEMENT outerRadius ( #PCDATA )>
<!ELEMENT startAngle ( #PCDATA )>
<!ELEMENT stopAngle ( #PCDATA )>
<!ELEMENT levelOfConfidence ( #PCDATA )>

<!-- Map reply -->
<!ELEMENT map_reply ( href | ( status_code, status_message,
                               status_code_extended? ) ) >
<!ATTLIST map_reply transaction_id ID #REQUIRED>
<!ELEMENT href ( #PCDATA )>


<!-- POI info request -->
<!-- The search_item may be a street or a pointofinterest item -->
<!-- If search_item is a street then all POIs on that street is returned-->
<!-- If search_item is a pointofinterest then info for it is returned -->
<!ELEMENT poi_info_request ( search_item, language ) >
<!ATTLIST poi_info_request transaction_id ID #REQUIRED
                           position_system %position_system_t; "MC2"
                           include_category_id %bool; "false"
                           include_full_search_item %bool; "false"
                           use_persistent_ids %bool; "false" >


<!-- POI info reply -->
<!ELEMENT poi_info_reply ( info_item* | ( status_code, status_message,
                                          status_code_extended? ) ) >
<!ATTLIST poi_info_reply transaction_id ID #REQUIRED>
<!ELEMENT info_item ( typeName, itemName, lat?, lon?, category_list?,
                      info_field*, search_item? )>
<!ATTLIST info_item numberfields %number; #REQUIRED
                    heading %number; #IMPLIED >
<!ELEMENT typeName ( #PCDATA )>
<!ELEMENT itemName ( #PCDATA )>
<!ELEMENT info_field ( fieldName, fieldValue ) >
<!ATTLIST info_field info_type %poi_info_t; #IMPLIED >
```

```
<!ELEMENT fieldName ( #PCDATA )>
<!ELEMENT fieldValue ( #PCDATA )>

<!-- simple poi desc request -->
<!ELEMENT simple_poi_desc_request EMPTY >
<!ATTLIST simple_poi_desc_request transaction_id ID #REQUIRED
                                  crc %hex_t; #REQUIRED >

<!ENTITY % te_t "(identity|base64)" >

<!-- crc_ok -->
<!ELEMENT crc_ok EMPTY >

<!-- simple poi desc reply -->
<!ELEMENT simple_poi_desc_reply ( simple_poi_desc_data | crc_ok ) >
<!ATTLIST simple_poi_desc_reply transaction_id ID #REQUIRED
                                crc %hex_t; #REQUIRED  >


<!ELEMENT simple_poi_desc_data (#PCDATA) >
<!ATTLIST simple_poi_desc_data te %te_t; #REQUIRED >

<!-- Send email request -->
<!ELEMENT email_request ( email_request_header,
                          (route_message_data | local_map_data | invite_email ) ) >
<!ATTLIST email_request transaction_id ID #REQUIRED>
<!ELEMENT invite_email EMPTY>
<!ELEMENT email_request_header ( email_address, subject,
                                 return_email_address? ) >
<!ATTLIST email_request_header
                           image_format %route_image_format_t; "png"
                           message_type %message_t; "html"
                           route_turn_image_type %route_turn_image_t; "map"
                           max_message_size %size_t; "inf"
                           overview_image_width    %size_t; #IMPLIED
                           overview_image_height   %size_t; #IMPLIED
                           route_turn_image_width  %size_t; #IMPLIED
                           route_turn_image_height %size_t; #IMPLIED
                           abbreviate_route_names  %bool;   #IMPLIED
                           route_landmarks         %bool;   #IMPLIED
                           route_only_overview     %bool;   #IMPLIED >
<!ELEMENT subject ( #PCDATA )>
<!ELEMENT return_email_address ( #PCDATA )>
<!-- Local map data -->
<!ELEMENT local_map_data ( language, signature, boundingbox,
                           local_map_string, map_symbol_list ) >
<!ELEMENT local_map_string ( #PCDATA ) >


<!-- Send email reply -->
```

```
<!ELEMENT email_reply ( status_code, status_message,
                        status_code_extended? ) >
<!ATTLIST email_reply transaction_id ID #REQUIRED>


<!-- SMS Format request -->
<!ENTITY % sms_version_t "(vicinity|eventfinder)" >
<!ELEMENT invite_sms (name) >
<!ATTLIST invite_sms type %sms_version_t; #REQUIRED >
<!ELEMENT place_sms (position_item|(search_item,language)) >
<!ATTLIST place_sms type %sms_version_t; #REQUIRED >

<!ELEMENT sms_format_request ( (smsmessage, phone_manufacturer, phone_model) |
                               (route_sms_message, route_message_data) |
                               (wayfinder_destination_sms) |
                               (wayfinder_route_sms) |
                               (wayfinder_favourite_sms) | invite_sms | place_sms ) >
<!ATTLIST sms_format_request transaction_id ID #REQUIRED
                             wayfinder_sms_version CDATA "1">


<!-- SMS Format reply -->
<!ELEMENT sms_format_reply ( (status_code, status_message,
                              status_code_extended?) |
                             sms_list )>
<!ATTLIST sms_format_reply transaction_id ID #REQUIRED>
<!ELEMENT sms_list ( smsmessage+ )>


<!-- Sort dist request -->
<!ELEMENT sort_dist_request ( (position_item | search_item),
                              (routeable_item_list | all_favorites) ) >
<!ATTLIST sort_dist_request
                         transaction_id ID #REQUIRED
                         max_number_reply_items %number; "1"
                         sort_distance %sort_distance_t; "radius"
                         route_cost %route_cost_t; "time"
                         position_system %position_system_t; "MC2"
                         route_vehicle %route_vehicle_t; "passengercar" >
<!ELEMENT all_favorites ( (user_id |
                           (user_session_id, user_session_key) ) )>


<!-- Sort dist reply -->
<!ELEMENT sort_dist_reply ( (sort_dist_list) |
                            ( status_code, status_message,
                              status_code_extended? ) ) >
<!ATTLIST sort_dist_reply transaction_id ID #REQUIRED>
<!ELEMENT sort_dist_list ( sort_dist_item+ )>
<!ELEMENT sort_dist_item ( (position_item | search_item | favorite) ) >
```

```
<!ATTLIST sort_dist_item
                     distance %number; #REQUIRED
                     estimated_time %number; #IMPLIED>


<!-- Top region request -->
<!ELEMENT top_region_request ( top_region_request_header ) >
<!ATTLIST top_region_request transaction_id ID     #REQUIRED
                             top_region_crc CDATA  #IMPLIED >
<!ELEMENT top_region_request_header ( language ) >
<!ATTLIST top_region_request_header
                    position_system %position_system_t; "MC2"
                    country             %bool; "true"
                    state               %bool; "false"
                    internationalRegion %bool; "false"
                    metaregion          %bool; "false" >


<!-- Top region reply -->
<!ELEMENT top_region_reply ( top_region_list | top_region_crc_ok |
                             ( status_code, status_message,
                               status_code_extended? ) ) >
<!ATTLIST top_region_reply transaction_id ID     #REQUIRED
                           top_region_crc CDATA  #IMPLIED >
<!ELEMENT top_region_list ( top_region* ) >
<!ATTLIST top_region_list numberitems %number; #REQUIRED >
<!ELEMENT top_region_crc_ok EMPTY >


<!-- Phone manufacturer request -->
<!ELEMENT phone_manufacturer_request EMPTY >
<!ATTLIST phone_manufacturer_request transaction_id ID #REQUIRED>


<!-- Phone manufacturer reply -->
<!ELEMENT phone_manufacturer_reply ( phone_manufacturer_list |
                                     ( status_code, status_message,
                                       status_code_extended? ) ) >
<!ATTLIST phone_manufacturer_reply transaction_id ID #REQUIRED>
<!ELEMENT phone_manufacturer_list ( phone_manufacturer* )>


<!-- Phone model request -->
<!ELEMENT phone_model_request ( phone_manufacturer? ) >
<!ATTLIST phone_model_request transaction_id ID #REQUIRED>


<!-- Phone model reply -->
<!ELEMENT phone_model_reply ( phone_model_list |
                              ( status_code, status_message,
```

```
                                status_code_extended? ) ) >
<!ATTLIST phone_model_reply transaction_id ID #REQUIRED>
<!ELEMENT phone_model_list ( phone_model* )>


<!-- User track request -->
<!ELEMENT user_track_request ( user_id | uin |
                               (user_session_id, user_session_key) ) >
<!ATTLIST user_track_request transaction_id ID #REQUIRED
                             start_time %time_t; #IMPLIED
                             end_time %time_t; #IMPLIED
                             max_nbr_tracks %size_t; "1"
                             position_system %position_system_t; "MC2" >


<!-- User track reply -->
<!ELEMENT user_track_reply ( ( user_track_item* ) |
                             ( status_code, status_message,
                               status_code_extended? ) ) >
<!ATTLIST user_track_reply transaction_id ID #REQUIRED>
<!ELEMENT user_track_item ( position_item ) >
<!ATTLIST user_track_item
                      time    %time_t;     #REQUIRED
                      dist    %number;     #IMPLIED
                      speed   %number;     #IMPLIED
                      source  CDATA        #REQUIRED >


<!-- User track add request -->
<!ELEMENT user_track_add_request ( ( user_id | uin |
                                     (user_session_id, user_session_key) ),
                                   user_track_item+ ) >
<!ATTLIST user_track_add_request transaction_id ID #REQUIRED >


<!-- User track add reply -->
<!ELEMENT user_track_add_reply ( status_code, status_message,
                                 status_code_extended? ) >
<!ATTLIST user_track_add_reply transaction_id ID #REQUIRED >


<!-- User debit log request -->
<!ELEMENT user_debit_log_request ( user_id |
                                   (user_session_id, user_session_key) ) >
<!ATTLIST user_debit_log_request
                                   transaction_id ID       #REQUIRED
                                   start_time     %time_t; #REQUIRED
                                   end_time       %time_t; #REQUIRED
                                   start_index    %size_t; "0"
                                   end_index      %size_t; "99" >
```

```
<!-- User debit log reply -->
<!ELEMENT user_debit_log_reply ( ( user_debit_log_element* ) |
                                 ( status_code, status_message,
                                   status_code_extended? ) ) >
<!ATTLIST user_debit_log_reply transaction_id ID       #REQUIRED
                               start_index    %size_t; #REQUIRED
                               end_index      %size_t; #REQUIRED
                               total_number_elements %size_t; #REQUIRED >
<!ELEMENT user_debit_log_element EMPTY >
<!ATTLIST user_debit_log_element
                                  message_id    %number; #REQUIRED
                                  debit_info    %number; #REQUIRED
                                  time          %time_t; #REQUIRED
                                  operationType %number; #REQUIRED
                                  sentSize      %size_t; #REQUIRED
                                  userOrigin    CDATA    #REQUIRED
                                  serverID      CDATA    #REQUIRED
                                  description   CDATA    #REQUIRED >


<!-- User find request -->
<!ELEMENT user_find_request ( user ) >
<!ATTLIST user_find_request transaction_id ID #REQUIRED >


<!-- User find reply -->
<!ELEMENT user_find_reply ( (user_id,uin)* |
                            ( status_code, status_message,
                              status_code_extended? ) ) >
<!ATTLIST user_find_reply transaction_id ID #REQUIRED >
<!ELEMENT uin (#PCDATA)>


<!-- Transactions request -->
<!ELEMENT transactions_request ( (user_id |
                                  (user_session_id,
                                   user_session_key))? ) >
<!ATTLIST transactions_request
                       transaction_id ID #REQUIRED
                       uin CDATA #IMPLIED
                       transaction_change %number; #IMPLIED >


<!-- Transactions reply -->
<!ELEMENT transactions_reply ( status_code, status_message,
                               status_code_extended? ) >
<!ATTLIST transactions_reply transaction_id ID #REQUIRED
                             nbr_transactions %number; #IMPLIED >
```

```
<!-- Transaction days request -->
<!ELEMENT transaction_days_request (
   (user_id | (user_session_id, user_session_key))? ) >
<!ATTLIST transaction_days_request
                      transaction_id     ID       #REQUIRED
                      uin                CDATA    #IMPLIED
                      check              %bool;   #IMPLIED
                      transaction_change %number; #IMPLIED >


<!-- Transaction days reply -->
<!ELEMENT transaction_days_reply ( status_code, status_message,
                                   status_code_extended? ) >
<!ATTLIST transaction_days_reply
                      transaction_id       ID       #REQUIRED
                      nbr_transaction_days %number; #REQUIRED
                      current_day          %time_t; #REQUIRED >


<!-- Activation request -->
<!ELEMENT activate_request ( phone_number?, new_password?,
                             name?, email?, opt_in?,
                             ( external_auth | server_auth_bob |
                               handle_me | hardware_id | hardware_key+ )? ) >
<!ATTLIST activate_request      transaction_id     ID       #REQUIRED
                                activation_code    CDATA    #IMPLIED
                                uin                CDATA    #IMPLIED
                                may_use            %bool;   "true"
                                create_new_token   %bool;   "true"
                                top_region_id      %number; #IMPLIED >
<!ELEMENT email ( #PCDATA ) >
<!ELEMENT opt_in EMPTY>
<!ATTLIST opt_in name CDATA #REQUIRED> <!-- prod-info -->
<!ELEMENT external_auth EMPTY>
<!ATTLIST external_auth type CDATA #REQUIRED>
<!ELEMENT handle_me ( licence_key? ) >
<!ATTLIST handle_me >
<!ELEMENT licence_key EMPTY>
<!ATTLIST licence_key key CDATA #REQUIRED >

<!ELEMENT hardware_id ( #PCDATA ) >
<!ATTLIST hardware_id type %hardware_key_type_t; #REQUIRED >

<!ELEMENT hardware_key ( #PCDATA ) >
<!ATTLIST hardware_key type %hardware_key_type_t; #REQUIRED >
```

```
<!-- Activation reply -->
<!ELEMENT activate_reply ( ( user_id?, auth_token? ) |
                           ( status_code, status_message,
                             status_code_extended? ) ) >
<!ATTLIST activate_reply  transaction_id       ID       #REQUIRED
                          uin                 CDATA    #IMPLIED >
<!ELEMENT auth_token ( #PCDATA ) >

<!-- External service request -->
<!ELEMENT ext_services_request EMPTY>
<!ATTLIST ext_services_request  transaction_id     ID           #REQUIRED
                                crc                CDATA        #IMPLIED
language           CDATA        #REQUIRED
                                new_client         %bool; "false" >

<!-- Extservice node -->
<!ELEMENT ext_service ( name, ext_service_help, icons, field* ) >
<!ATTLIST ext_service          background_colour    CDATA       #REQUIRED
                               service_id          %number;    #REQUIRED
                               type                CDATA       #REQUIRED >

<!-- Extservice field node -->
<!ELEMENT field ( field_name, field_option* ) >
<!ATTLIST field                id                   %number;    #REQUIRED
                               type                 CDATA       #REQUIRED
                               req                  %hex_t;     #REQUIRED
                               nbr_choices          %number;    #IMPLIED >

<!-- Extservice field name -->
<!ELEMENT field_name ( #PCDATA ) >
<!-- Extservice field option -->
<!ELEMENT field_option ( field_option_name ) >
<!ATTLIST field_option         id                   %number;    #REQUIRED >
<!-- Extservice field option name -->
<!ELEMENT field_option_name ( #PCDATA ) >
<!-- Extservice help text -->
<!ELEMENT ext_service_help ( #PCDATA ) >
<!-- Extservice background colour -->
<!ELEMENT background_colour ( #PCDATA ) >
<!-- Icons -->
<!ELEMENT icons ( icon* ) >
<!ATTLIST icons nbr_icons %number; #REQUIRED >
<!-- Icon -->
<!ELEMENT icon EMPTY >
<!ATTLIST icon name        CDATA        #REQUIRED
               xsize       %number;     #REQUIRED
               ysize       %number;     #REQUIRED
               client_type CDATA        #REQUIRED >

<!-- External services reply -->
```

```
<!ELEMENT ext_services_reply ( ext_services_crc_ok |  ext_service* |
                              ( status_code, status_message,
                                status_code_extended? ) ) >
<!ATTLIST ext_services_reply   crc                  CDATA       #REQUIRED
                               transaction_id       ID          #REQUIRED
                               nbr_services         %number;    #REQUIRED >

<!-- External services crc ok -->
<!ELEMENT ext_services_crc_ok EMPTY >

<!-- External search request -->
<!ELEMENT ext_search_request ( field_val* ) >
<!ATTLIST ext_search_request   search_item_starting_index %number; #REQUIRED
                               search_item_ending_index   %number; #REQUIRED
                               service_id                 %number; #REQUIRED
                               language                   CDATA    #REQUIRED
                               transaction_id             ID       #REQUIRED >

<!-- External search request field val -->
<!ELEMENT field_val ( #PCDATA ) >
<!ATTLIST field_val            id                         %number; #REQUIRED >


<!-- Tunnel request -->
<!ELEMENT tunnel_request ( post_data? ) >
<!ATTLIST tunnel_request   transaction_id     ID       #REQUIRED
                           url                %HREF;   #REQUIRED  >
<!ELEMENT post_data ( #PCDATA ) >

<!ATTLIST post_data te %te_t; "identity">

<!-- Tunnel reply -->
<!ELEMENT tunnel_reply ( ( header*, body? ) |
                           ( status_code, status_message,
                             status_code_extended? ) ) >
<!ATTLIST tunnel_reply  transaction_id       ID       #REQUIRED
                        status_line         CDATA    #IMPLIED >

<!-- Zoom Settings request -->
<!ELEMENT zoom_settings_request EMPTY >
<!ATTLIST zoom_settings_request transaction_id ID #REQUIRED
                                crc %hex_t; #REQUIRED
                                pixel_size %number; #IMPLIED >

<!-- Zoom Settings reply -->
<!ELEMENT zoom_settings_reply (zoom_levels | crc_ok ) >
<!ATTLIST zoom_settings_reply transaction_id ID #REQUIRED >
<!ELEMENT zoom_levels (zoom_level+)>
<!ATTLIST zoom_levels crc %hex_t; #REQUIRED
                      nbr_zoom_levels %number; #REQUIRED
```

```
                      pixel_size %number; #REQUIRED>
<!ELEMENT zoom_level EMPTY>
<!ATTLIST zoom_level max_x %number; #REQUIRED
                     max_y %number; #REQUIRED
                     min_x %number; #REQUIRED
                     min_y %number; #REQUIRED
                     zoom_level_nbr %number; #REQUIRED
                     zoom_j2me %bool; #IMPLIED>

<!ELEMENT header EMPTY>
<!ATTLIST header field CDATA #REQUIRED
                 value CDATA #REQUIRED >
<!ELEMENT body ( #PCDATA ) >
<!ATTLIST body te %te_t; "base64">


<!-- Error report -->
<!ELEMENT error_report ( error_message )>
<!ATTLIST error_report transaction_id     ID       #REQUIRED
                       subject            CDATA    #IMPLIED >
<!ELEMENT error_message ( #PCDATA ) >

<!-- Error report reply -->
<!ELEMENT error_report_reply ( status_code, status_message,
                               status_code_extended? ) >
<!ATTLIST error_report_reply transaction_id     ID       #REQUIRED>

<!-- POI review -->
<!ENTITY % poi_grade "(0|1|2|3|4|5)" >
<!ELEMENT poi_review_requests ( poi_review_add_request |
                                poi_review_delete_request |
                                poi_review_list_request )+ >
<!ELEMENT poi_review_replies  ( poi_review_add_reply |
                                poi_review_delete_reply |
                                poi_review_list_reply )+ >

<!ELEMENT poi_review_add_request ( ( user_id | uin |
                                     ( user_session_id, user_session_key ) )? ,
                                   ( poi_review_title, poi_review_text )? ) >
<!ATTLIST poi_review_add_request transaction_id  ID           #REQUIRED
                                 poi_id          CDATA        #REQUIRED
                                 grade           %poi_grade;  #REQUIRED
                                 lang            %language_t; #IMPLIED
                                 review_id       CDATA        #IMPLIED >

<!ELEMENT poi_review_add_reply ( status_code, status_message,
                                 status_code_extended? )? >
<!ATTLIST poi_review_add_reply transaction_id  ID     #REQUIRED
                               review_id       CDATA  #IMPLIED >
```

```
<!ELEMENT poi_review_delete_request ( user_id | uin |
                                    ( user_session_id, user_session_key ) )? >
<!ATTLIST poi_review_delete_request transaction_id  ID     #REQUIRED
                                    review_id       CDATA  #REQUIRED >

<!ELEMENT poi_review_delete_reply ( status_code, status_message,
                                    status_code_extended? )? >
<!ATTLIST poi_review_delete_reply transaction_id  ID      #REQUIRED >

<!ELEMENT poi_review_list_request ( user_id | uin |
                                    ( user_session_id, user_session_key ) |
                                    poi_review_poi | poi_review_id ) >
<!ENTITY % poi_review_details "(none|some|all)" >
<!ATTLIST poi_review_list_request transaction_id  ID                   #REQUIRED
                                  details         %poi_review_details; "all"
                                  lang            %language_t;         #IMPLIED >

<!ELEMENT poi_review_poi EMPTY >
<!ATTLIST poi_review_poi poi_id CDATA #REQUIRED >
<!ELEMENT poi_review_id  EMPTY >
<!ATTLIST poi_review_id  review_id CDATA #REQUIRED >

<!ELEMENT poi_review_list_reply ( ( poi_review* ) |
                                  ( status_code, status_message,
                                    status_code_extended? ) ) >
<!ATTLIST poi_review_list_reply transaction_id  ID      #REQUIRED >

<!ELEMENT poi_review ( poi_review_detail* ) >
<!ATTLIST poi_review poi_id      CDATA        #REQUIRED
                    avg_grade   CDATA        #IMPLIED
                    grade_count %number;     #IMPLIED >

<!ELEMENT poi_review_detail ( uin?, poi_review_title?, poi_review_text? ) >
<!ATTLIST poi_review_detail review_id  CDATA       #REQUIRED
                            date       CDATA       #REQUIRED
                            grade      %poi_grade; #IMPLIED
                            logonID    CDATA       #IMPLIED
                            firstname  CDATA       #IMPLIED
                            lastname   CDATA       #IMPLIED >

<!ELEMENT poi_review_title ( #PCDATA ) >
<!ELEMENT poi_review_text ( #PCDATA ) >

<!-- Show activation code -->
<!ELEMENT show_activationcode_request EMPTY >
<!ATTLIST show_activationcode_request transaction_id ID #REQUIRED
                                      actvationcode  CDATA #REQUIRED>
<!ELEMENT show_activationcode_reply ( (status_code, status_message,
                                       status_code_extended?)? ) >
<!ATTLIST show_activationcode_reply transaction_id ID #REQUIRED
```

```
                                    rights   CDATA    #IMPLIED
                                    server   CDATA    #IMPLIED
                                    ownerUIN %number; #IMPLIED >


<!-- Expand top region -->
<!ELEMENT expand_top_region_request ( top_region* ) >
<!ATTLIST expand_top_region_request
              transaction_id      ID           #REQUIRED
              language            %language_t; #IMPLIED
              position_system     %position_system_t; "MC2"
              country             %bool;       "true"
              state               %bool;       "false"
              internationalRegion %bool;       "false"
              metaregion          %bool;       "false"  >
<!ELEMENT expand_top_region_reply ( top_region_list* |
                                    (status_code, status_message,
                                     status_code_extended?) ) >
<!ATTLIST expand_top_region_reply transaction_id ID #REQUIRED >


<!-- Get client type info -->
<!ELEMENT client_type_info_request EMPTY >
<!ATTLIST client_type_info_request
   transaction_id      ID           #REQUIRED
   client_type         CDATA        #REQUIRED
   client_type_options CDATA        #REQUIRED >

<!ELEMENT client_type_info_reply ( (status_code, status_message,
                                    status_code_extended?)? ) >
<!ATTLIST client_type_info_reply
   transaction_id      ID           #REQUIRED
   phoneModel          CDATA        #REQUIRED
   imageExtension      CDATA        #REQUIRED
   extraRights         CDATA        #REQUIRED >


<!-- Get server list for client type and uin -->
<!ELEMENT server_list_for_client_type_request EMPTY >
<!ATTLIST server_list_for_client_type_request
   transaction_id      ID           #REQUIRED
   client_type         CDATA        #REQUIRED
   client_type_options CDATA        #REQUIRED
   srvt                CDATA        #REQUIRED
   uin                 CDATA        #IMPLIED >

<!ELEMENT server_list_for_client_type_reply ( server_list |
                                              (status_code, status_message,
                                               status_code_extended?) ) >
<!ATTLIST server_list_for_client_type_reply
```

```
   transaction_id      ID           #REQUIRED >

<!-- Create wayfinder user -->
<!ELEMENT create_wayfinder_user_request ( hardware_key+ ) >
<!ATTLIST create_wayfinder_user_request
   transaction_id      ID           #REQUIRED
   client_type         CDATA        #REQUIRED
   client_type_options CDATA        #REQUIRED
   client_lang         %language_t; #REQUIRED
   logon               CDATA        #REQUIRED
   password            CDATA        #REQUIRED
   activation_code     CDATA        #IMPLIED
   top_region_id       %number;     #IMPLIED >

<!ELEMENT create_wayfinder_user_reply ( (status_code, status_message,
                                         status_code_extended?,
                                         server_list?)? ) >
<!ATTLIST create_wayfinder_user_reply transaction_id ID    #REQUIRED
                                      uin            CDATA #IMPLIED >


<!-- Update user's hardware keys -->
<!ELEMENT update_hardware_keys_request ( hardware_key+ ) >
<!ATTLIST update_hardware_keys_request
   transaction_id      ID           #REQUIRED
   uin                 CDATA        #IMPLIED
   client_type         CDATA        #IMPLIED
   client_type_options CDATA        #IMPLIED >

<!ELEMENT update_hardware_keys_reply ( (status_code, status_message,
                                        status_code_extended?)? ) >
<!ATTLIST update_hardware_keys_reply transaction_id ID #REQUIRED >

<!-- Get stored user data request -->
<!ELEMENT get_stored_user_data_request EMPTY >
<!ATTLIST get_stored_user_data_request
   transaction_id       ID          #REQUIRED
   uin                  CDATA       #REQUIRED
   key                  CDATA       #IMPLIED >

<!-- Get stored user data reply -->
<!ELEMENT get_stored_user_data_reply ( ( stored_user_data ) |
            ( status_code, status_message, status_code_extended? ) ) >
<!ATTLIST get_stored_user_data_reply
   transaction_id       ID          #REQUIRED >
<!ELEMENT stored_user_data EMPTY >
<!ATTLIST stored_user_data
   key                  CDATA       #REQUIRED
   value                CDATA       #REQUIRED >
```

```
<!-- Set stored user data  request -->
<!ELEMENT set_stored_user_data_request ( stored_user_data ) >
<!ATTLIST set_stored_user_data_request
   transaction_id       ID          #REQUIRED
   uin                  CDATA       #REQUIRED >

<!-- Set stored user data reply -->
<!ELEMENT set_stored_user_data_reply ( ( status_code, status_message,
                                         status_code_extended? )? ) >
<!ATTLIST set_stored_user_data_reply
   transaction_id       ID          #REQUIRED >


<!-- Friend finder, request from client to update position and get friends list -->
<!ELEMENT friend_finder_request EMPTY >
<!ATTLIST friend_finder_request
   position_system          %position_system_t;   "MC2"
   lat                      %number;              #IMPLIED
   lon                      %number;              #IMPLIED
   transaction_id           ID                    #REQUIRED >

<!ELEMENT friend_finder_reply ( friend_list |
                                (status_code, status_message,
                                 status_code_extended?) ) >
<!ATTLIST friend_finder_reply   transaction_id ID   #REQUIRED >
<!ELEMENT friend_list ( friend* ) >
<!ATTLIST friend_list nbr   %number;              #REQUIRED >
<!ELEMENT friend EMPTY >
<!ATTLIST friend
                uin                     CDATA             #REQUIRED
                name                    CDATA             #REQUIRED
                status                  CDATA             #REQUIRED
                description             CDATA             #IMPLIED
                avatar_image_name       CDATA             #REQUIRED
                thumb_icon_name         CDATA             #REQUIRED
                phone_number            CDATA             #REQUIRED
                email_address           CDATA             #REQUIRED
                lat                     %number;          #REQUIRED
                lon                     %number;          #REQUIRED
                location                CDATA             #REQUIRED
                time_stamp              %time_t;          #REQUIRED >


<!-- Information about a user's friend -->
<!ELEMENT friend_finder_info_request EMPTY >
<!ATTLIST friend_finder_info_request  uin             CDATA   #REQUIRED
                                      friend_uin      CDATA   #REQUIRED
                                      position_system %position_system_t; "MC2"
                                      transaction_id  ID      #REQUIRED >
```

```
<!ELEMENT friend_finder_info_reply ( friend |
                                     (status_code, status_message,
                                      status_code_extended?) ) >
<!ATTLIST friend_finder_info_reply   transaction_id ID   #REQUIRED >


<!-- Cell ID request -->
<!ELEMENT TGPP EMPTY >
<!ATTLIST TGPP
   c_mcc             CDATA #IMPLIED
   c_mnc             CDATA #IMPLIED
   lac               CDATA #IMPLIED
   cell_id           CDATA #IMPLIED
   network_type      CDATA #IMPLIED
   signal_strength   CDATA #IMPLIED >
<!ELEMENT CDMA EMPTY >
<!ATTLIST CDMA
   sid               CDATA #IMPLIED
   nid               CDATA #IMPLIED
   bid               CDATA #IMPLIED
   network_type      CDATA #IMPLIED
   signal_strength   CDATA #IMPLIED >
<!ELEMENT iDEN EMPTY >
<!ATTLIST iDEN
   c_mcc             CDATA #IMPLIED
   dnc               CDATA #IMPLIED
   sa_id             CDATA #IMPLIED
   lla_id            CDATA #IMPLIED
   cell_id           CDATA #IMPLIED
   signal_strength   CDATA #IMPLIED >
<!ELEMENT cell_id_request (TGPP | CDMA | iDEN) >
<!ATTLIST cell_id_request
   position_system   %position_system_t;  "MC2"
   transaction_id    ID                   #REQUIRED >

<!-- Cell ID reply -->
<!ELEMENT cell_id_reply ( ( position_item ) |
            ( status_code, status_message, status_code_extended? ) ) >
<!ATTLIST cell_id_reply
   transaction_id    ID       #REQUIRED
   inner_radius      %number; #IMPLIED
   outer_radius      %number; #IMPLIED
   altitude          %number; #IMPLIED
   start_angle       %number; #IMPLIED
   end_angle         %number; #IMPLIED >


<!-- Local category tree request -->
<!ELEMENT local_category_tree_request ( position_item ) >
<!ATTLIST local_category_tree_request transaction_id ID #REQUIRED
```

```
                                    crc CDATA #REQUIRED
                                    language %language_t; #REQUIRED
                                    version %number; #REQUIRED >

<!-- Local category tree reply -->
<!ELEMENT category_table ( #PCDATA ) >
<!ATTLIST category_table length %number; #REQUIRED >
<!ELEMENT lookup_table ( #PCDATA ) >
<!ATTLIST lookup_table length %number; #REQUIRED >
<!ELEMENT string_table ( #PCDATA ) >
<!ATTLIST string_table length %number; #REQUIRED >
<!ELEMENT local_category_tree_reply ( (category_table, lookup_table,
                                    string_table)|crc_ok|( status_code,
                                    status_message, status_uri?,
                                    status_code_extended? ) ) >
<!ATTLIST local_category_tree_reply transaction_id ID #REQUIRED
                                    crc CDATA #IMPLIED >

<!-- POI detail request -->
<!ELEMENT poi_detail_request ( itemid ) >
<!ATTLIST poi_detail_request transaction_id ID           #REQUIRED
                             language       %language_t; #REQUIRED>

<!-- POI detail reply -->
<!ELEMENT poi_detail_reply ( ( detail_item, resources? ) |
  ( status_code, status_message,
  status_code_extended? ) ) >
<!ATTLIST poi_detail_reply transaction_id ID #REQUIRED>

<!ELEMENT detail_item ( detail_field* )>
<!ATTLIST detail_item numberfields %number; #REQUIRED >

<!ELEMENT detail_field ( fieldName, fieldValue ) >
<!ATTLIST detail_field detail_type %poi_detail_t; #IMPLIED
                       detail_content %poi_detail_content_t; #IMPLIED >

<!ELEMENT resources ( image_group*, review_group* ) >
<!ATTLIST resources number_image_groups %number; #REQUIRED
                    number_review_groups %number; #REQUIRED >

<!ELEMENT image_group ( image* ) >
<!ATTLIST image_group number_images %number; #REQUIRED
  provider_name CDATA #REQUIRED
  provider_image CDATA #REQUIRED >

<!ELEMENT image EMPTY >
<!ATTLIST image url CDATA #REQUIRED >

<!ELEMENT review_group ( review* ) >
<!ATTLIST review_group number_reviews %number; #REQUIRED
```

```
  provider_name CDATA #REQUIRED
  provider_image CDATA #REQUIRED >

<!ELEMENT review ( #PCDATA ) >
<!ATTLIST review rating %number; #REQUIRED
  date CDATA #REQUIRED
  reviewer CDATA #REQUIRED >

<!-- Search match -->

<!ENTITY % search_match_type_t "(street|pointofinterest|misc|person|
                                 other)">
<!ELEMENT search_match ( name, itemid,
                         location_name, lat?, lon?, category_list?,
                         search_area*, detail_item? )>
<!ATTLIST search_match search_match_type %search_match_type_t; #REQUIRED
                       category_image CDATA #REQUIRED
                       provider_image CDATA #REQUIRED
                       brand_image CDATA #REQUIRED
                       additional_info_exists %bool; #REQUIRED>

<!-- One list search request -->

<!-- The sorting of the resulting search_hit_list -->
<!ENTITY % sorting_t "(alfa_sort|distance_sort)">
<!ENTITY % search_for_type_t "(address|all)">

<!ELEMENT one_search_request ( search_match_query?,
                               category_list?,
                               ( ( position_item, distance? ) |
                               ( query_location, top_region_id ) ) ) >
<!ATTLIST one_search_request transaction_id    ID            #REQUIRED
                             max_number_matches %number;     #REQUIRED
                             language          %language_t;  #REQUIRED
                             round             %number;      #REQUIRED
                             version           %number;      #REQUIRED
                             include_detail_fields %bool;      #IMPLIED
                             position_system %position_system_t; "MC2"
                             sorting           %sorting_t;   #REQUIRED
                             search_type %search_for_type_t; "all" >
<!ELEMENT search_match_query ( #PCDATA )>
<!ELEMENT query_location ( #PCDATA )>

<!-- One list search reply -->
<!ELEMENT one_search_reply ( search_list |
                             ( status_code, status_message,
                               status_uri? ) ) >
<!ATTLIST one_search_reply transaction_id  ID      #REQUIRED >

<!ELEMENT search_list ( search_match* )>
```

```
<!ATTLIST search_list number_matches       %number; #REQUIRED
                      total_number_matches %number; #REQUIRED >

<!-- Server info request -->
<!ELEMENT server_info_request EMPTY>
<!ATTLIST server_info_request transaction_id ID #REQUIRED
                              client_type CDATA #REQUIRED
                              client_type_options CDATA #IMPLIED
                              client_version CDATA #REQUIRED >

<!-- Server info reply -->
<!ELEMENT server_info_reply ( client_type_info |
                            ( status_code, status_message,
                              status_uri? ) ) >
<!ATTLIST server_info_reply transaction_id ID #REQUIRED >

<!ELEMENT client_type_info EMPTY>
<!ATTLIST client_type_info upgrade_available %bool; #REQUIRED
                           latest_version CDATA #REQUIRED
                           force_upgrade %bool; #REQUIRED
                           upgrade_id CDATA #IMPLIED >
```

47 Direct image interface

When making image URIs the Map Request is recommended, see Section 17.1. But if you can't use Map Request and want to use images only here is a direct way to images.
Beware that this direct image interface might change in the future. We recommend using the Map Request if possible.
The URI is like Map.[ext]?[parameters] where [ext] is one of:

png Portable Network Graphics.

gif Graphics Interchange Format.

wbmp Wireless BitMaP, WAP image.

Other formats may be added in the future.

The parameters supported in [parameters] are:

lla Lower latitude, south latitude in MC2 units. Must be present.

llo Lower longitude, west longitude in MC2 units. Must be present.

ula Upper latitude, north latitude in MC2 units. Must be present.

ulo Upper longitude, east longitude in MC2 units. Must be present.

w Width of image in pixels. Must be present.

h Height of image in pixels. Must be present.

s If greater than 4096 then text in image. Road names, cities lakes etc. Must be present.

map 1 then map stuff, roads, parks etc, in image. 0 no map data in image.

topomap 1 then topograthical map data in image. NB! No topograthical data at present. 0 no topograthical map data in image.

poi 1 then points of interest in image. Symbols added to image. 0 no pois.

scale 1 then draw scale at lower right corner. 0 no scale.

traffic 1 then draw traffic information in image. Traffic information from Swedish Vägverket. 0 no traffic information.

mt Map type, allowed values are "std" and "wap". Images with wap setting has more contrast in them. Default is “std”.

ms Map symbol to draw paramerer. Multiple ms parameters are allowed to make it possible to add many map symbols. The parameter value is $1_$2_$3_$4_$5 where:

- **$1** Is type of symbol: 0 PIN, $5 ignored, 1 USER_DEFINED, $5 used.
- **$2** Is latitude for symbol in MC2 units
- **$3** Is longitude for symbol in MC2 units.
- **$4** Is name of symbol base64- and then url-encoded. Currently not used. Empty string is recommened.
- **$5** Is image to draw base64- and then url-encoded. Empty string is recommended with $1 = 0.

r Route ID, an id for a route to draw in image. See Section 11.2 for more information about route id.

route 1 then route identified by “r” parameter is shown. 0 disables route on map even if “r” parameter is present and valid.

turn The turn index, in hexadecimal with capital letters, to draw a turn arrow for. Start is 0 first turn is 1.

auth A authorization string received from Wayfinder Systems. Might be required in the future.

lang Language code as ISO-639.

sesi Session ID, used with Session Key.

si Alias for sesi.

sesk Session key.

sk Alias for sesk.

uin User Identification number, used with tok and hwd.

tok Token, from token authentication.

hwd Hardware key, used with hwdt to identify user.

hwdt Hardware key type.

An example pinpointing the location of Wayfinder Systems in Lund, Sweden.
Map.png?lla=664632208&
llo=157172530&ula=664832208&ulo=157527596&w=500&h=500&s=31250&mt=std&
map=1&topomap=1&poi=1&route=0&scale=1&traffic=1&ms=0_664732208_157350063__

48 Examples

Some examples of the usage of the API formally described above.

48.1 User Request Example

Please note that this example is quite unrealistic since each user is modified in different transactions within the same document. But it shows how it could have been done if the transactions was in different requests.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE isab-mc2 SYSTEM "isab-mc2.dtd">

<isab-mc2>

   <auth>
      <auth_user>login</auth_user>
      <auth_passwd>password</auth_passwd>
   </auth>

   <user_request transaction_id="ID1">
      <!-- Example showing the addition of user testuser -->
      <user>
         <user_id>testuser</user_id>
         <first_name>User</first_name>
         <last_name>Test</last_name>
         <initials>UB</initials>
         <default_transportation>Passenger car</default_transportation>
         <language>Swedish</language>
         <measurement_system>Metric</measurement_system>
         <service>
            <service_type>Route</service_type>
            <service_method>SMS</service_method>
         </service>
         <service>
            <service_type>Route</service_type>
            <service_method>WAP</service_method>
         </service>
         <phone>
```

```
            <phone_number>4623456789</phone_number>
            <phone_manufacturer>Ericsson</phone_manufacturer>
            <phone_model>R320S</phone_model>
         </phone>
      </user>
   </user_request>

   <user_request transaction_id="ID2">
      <!-- Example showing the addition of user test1 -->
      <user>
         <user_id>test1</user_id>
         <first_name>Test</first_name>
         <last_name>Name</last_name>
         <initials>TN</initials>
         <default_transportation>Pedestrian</default_transportation>
         <language>Swedish</language>
         <measurement_system>Metric</measurement_system>
         <service>
            <service_type>Route</service_type>
            <service_method>SMS</service_method>
         </service>
         <phone>
            <phone_number>46123456789</phone_number>
            <phone_manufacturer>Ericsson</phone_manufacturer>
            <phone_model>A1018S</phone_model>
         </phone>
      </user>
   </user_request>

   <user_request transaction_id="ID3">
      <!-- Example showing that the SMS-access for user test1
           is removed, and that the WAP-access is added -->
      <user>
         <user_id>test1</user_id>
         <service>
            <service_type>Route</service_type>
            <service_method>SMS</service_method>
            <service_delete />
         </service>
         <service>
            <service_type>Route</service_type>
            <service_method>WAP</service_method>
         </service>
      </user>
   </user_request>

   <user_request transaction_id="ID4">
      <!-- Example showing that user test1 has changed
           phone manufacturer and model -->
      <user>
```

```
         <user_id>test1</user_id>
         <phone>
            <phone_number>46123456789</phone_number>
            <phone_manufacturer>Nokia</phone_manufacturer>
            <phone_model>7110</phone_model>
         </phone>
      </user>
   </user_request>

   <user_request transaction_id="ID5">
      <!-- Example showing the change of language for user test1 -->
      <user>
         <user_id>test1</user_id>
         <language>English</language>
      </user>
   </user_request>

   <user_request transaction_id="ID6">
      <!-- Adding SMS-access for user testuser -->
      <user>
         <user_id>testuser</user_id>
         <service>
            <service_type>Route</service_type>
            <service_method>SMS</service_method>
         </service>
      </user>
   </user_request>

   <user_request transaction_id="ID7">
      <!-- Adding Cellular 46123456789 to testuser -->
      <user>
         <user_id>testuser</user_id>
 <phone>
            <phone_number>46123456789</phone_number>
         </phone>
      </user>
   </user_request>

</isab-mc2>
```

48.2 User Reply Example

An example of reply from the $\mathcal{MC}^2$-system to the example in *48.1 - User request example*.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE isab-mc2 SYSTEM "isab-mc2.dtd">

<isab-mc2>

   <!-- User testuser added alright-->
```

```
   <user_reply transaction_id="ID1">
      <status_code>0</status_code>
      <status_message>OK</status_message>
   </user_reply>

   <!-- User test1 added alright-->
   <user_reply transaction_id="ID2">
      <status_code>0</status_code>
      <status_message>OK</status_message>
   </user_reply>

   <!-- SMS-access for user test1 is removed, and WAP-access
           is added -->
   <user_reply transaction_id="ID3">
      <status_code>0</status_code>
      <status_message>OK</status_message>
   </user_reply>

   <!-- Manufacturer for user test1 changed alright -->
   <user_reply transaction_id="ID4">
      <status_code>0</status_code>
      <status_message>OK</status_message>
   </user_reply>

   <!-- Language for user test1 changed alright -->
   <user_reply transaction_id="ID5">
      <status_code>0</status_code>
      <status_message>OK</status_message>
   </user_reply>

   <!-- User testuser already had SMS-access -->
   <user_reply transaction_id="ID6">
      <status_code>0</status_code>
      <status_message>OK</status_message>
   </user_reply>

   <!-- User testuser couldn't add cellular 46123456789 -->
   <user_reply transaction_id="ID7">
      <status_code>-1</status_code>
      <status_message>Failed to change user.</status_message>
   </user_reply>

</isab-mc2>
```

48.3 Search, Route and Expand Request Example

A simple example of how to make a search, route or expand request.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE isab-mc2 SYSTEM "isab-mc2.dtd">
```

```
<isab-mc2>

   <auth>
      <auth_user>login</auth_user>
      <auth_passwd>pass</auth_passwd>
   </auth>

   <search_request transaction_id="IDSearch1">
      <search_request_header>
         <search_preferences>
            <search_settings matchtype="close"
                             wordmatch="beginning"
                             sorttype="confidence_sort" >
               <search_for_municipal/>
               <search_for_city/>
               <search_for_citypart/>
               <search_for_street/>
               <search_for_company/>
            </search_settings>
         </search_preferences>
      </search_request_header>

      <!-- Search query -->
      <search_query>
         <search_area_query>Lund</search_area_query>
         <search_item_query>itine</search_item_query>
      </search_query>
   </search_request>


   <search_request transaction_id="IDSearch2">
      <search_request_header>
         <search_preferences>
    <user_id>testuser</user_id>
         </search_preferences>
      </search_request_header>

      <!-- Search query -->
      <search_query>
         <search_area search_area_type="city">
            <name>MALMÖ</name>
            <areaid>0;33;33000034</areaid>
         </search_area>
         <search_item_query>Brog11</search_item_query>
      </search_query>
   </search_request>

   <route_request transaction_id="IDRoute1">
      <route_request_header>
         <route_preferences route_description_type="normal">
```

```
            <user_id>testuser</user_id>
         </route_preferences>
      </route_request_header>
      <!-- Origin -->
      <routeable_item_list>
         <position_item position_system="WGS84">
            <lat>N 552312</lat>
            <lon>E 131256</lon>
         </position_item>
      </routeable_item_list>

      <!-- Destination -->
      <routeable_item_list>
         <search_item search_item_type="pointofinterest">
            <name>Itinerary Systems IS AB</name>
            <itemid>0;70007D86;0</itemid>
         </search_item>
      </routeable_item_list>
   </route_request>


   <expand_request transaction_id="IDExpand1">
      <expand_request_header>
         <search_preferences>
            <user_id>testuser</user_id>
         </search_preferences>
      </expand_request_header>
      <expand_request_query>
         <search_area search_area_type="city">
            <name>MALMÖ</name>
            <areaid>0;33;33000034</areaid>
         </search_area>
         <search_item search_item_type="category">
            <name>Restauranger</name>
            <itemid>0;78000015;0</itemid>
         </search_item>
      </expand_request_query>
   </expand_request>
</isab-mc2>
```

48.4 Search, Route and Expand Reply Example

An example of reply from the $\mathcal{MC}^2$ 2-system to the examples in 48.3 - Search, route and expand request.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE isab-mc2 SYSTEM "isab-mc2.dtd">

<isab-mc2>

   <search_reply transaction_id="ISSearch1">
```

```
      <!-- Search reply -->
      <search_area_list numberitems="1">
         <search_area search_area_type="city">
            <name>LUND</name>
            <areaid>0;300A67EA;30</areaid>
         </search_area>
      </search_area_list>
      <search_item_list numberitems="1">
         <search_item search_item_type="pointofinterest">
            <name>Itinerary Systems IS AB</name>
            <itemid>0;280CB45;120</itemid>
         </search_item>
      </search_item_list>
   </search_reply>

   <search_reply transaction_id="ISSearch2">
      <!-- Search reply -->
      <search_area_list numberitems="1">
         <search_area search_area_type="city">
            <name>MALMÖ</name>
            <areaid>0;300A67EF;31</areaid>
         </search_area>
      </search_area_list>
      <search_item_list numberitems="2">
         <search_item search_item_type="street">
            <name>Brogatan</name>
            <streetnbr>11</streetnbr>
            <itemid>0;300CA57;20</itemid>
         </search_item>
         <search_item search_item_type="street">
            <name>Brovägen</name>
            <streetnbr>11</streetnbr>
            <itemid>0;3005671;26</itemid>
         </search_item>
      </search_item_list>
   </search_reply>

   <route_reply transaction_id="IDRoute1">
      <route_reply_header>
         <total_distance>21 km</total_distance>
         <total_time>16 min 45 s</total_time>
         <total_standstilltime>55 s</total_standstilltime>
         <average_speed>75 km/h</average_speed>
         <routing_vehicle>Passenger car</routing_vehicle>
      </route_reply_header>

      <route_origin>
         <search_item>
            <name>Amiralsgatan</name>
            <streetnbr>56</streetnbr>
```

```
            <itemid>0;3007643;75</itemid>
         </search_item>
      </route_origin>
      <route_destination>
         <search_item>
            <name>Itinerary Systems IS AB</name>
            <itemid>0;280CB45;120</itemid>
         </search_item>
      </route_destination>

      <route_reply_items>
         <route_reply_item>
            <description>Starta nordost vid Amiralsg</description>
         </route_reply_item>
         <route_reply_item>
            <description>Kör 39 meter sväng sen vänster in på den 1:a vägen Amiralsg</
         </route_reply_item>
         <route_reply_item>
            <description>Kör 280 meter sväng sen höger in på den 1:a vägen Drottningg<
         </route_reply_item>
         <route_reply_item>
            <description>Kör 1.3 km kör sen in på E6</description>
         </route_reply_item>
         <route_reply_item>
            <description>Kör 400 meter sväng sen vänster in på den 1:a vägen Barav</des
         </route_reply_item>
         <route_reply_item>
            <description>Kör slutligen 160 meter fram till Barav 1</description>
         </route_reply_item>
      </route_reply_items>
   </route_reply>

   <expand_reply transaction_id="IDExpand1">
      <search_item_list numberitems="3">
         <search_item search_item_type="company">
            <name>Blue Diamond</name>
            <itemid>0;220518;56</itemid>
         </search_item>
         <search_item search_item_type="company">
            <name>Acapullo</name>
            <itemid>0;220519;56</itemid>
         </search_item>
         <search_item search_item_type="company">
            <name>Kaffehuset</name>
            <itemid>0;220520;56</itemid>
         </search_item>
      </search_item_list>
   </expand_reply>

</isab-mc2>
```

48.5 Simple Search Request Example

An example of a search request to send to the $\mathcal{MC}^2$ 2-system.

```
<?xml version='1.0' encoding='iso-8859-1' ?>
<!DOCTYPE isab-mc2 SYSTEM 'isab-mc2.dtd'>

<isab-mc2>

   <auth>
      <auth_user>user</auth_user>
      <auth_passwd>pass</auth_passwd>
   </auth>

   <search_request transaction_id="IDSearch1">
      <search_request_header>
         <search_preferences>
            <search_settings matchtype="close"
                             wordmatch="beginning"
                             sorttype="confidence_sort" >
               <search_for_municipal/>
               <search_for_city/>
               <search_for_citypart/>
               <search_for_street/>
               <search_for_company/>
     <search_for_category/>
     <search_for_misc/>
          </search_settings>
        </search_preferences>
      </search_request_header>
      <search_query>
         <search_area_query>Stockholm</search_area_query>
         <search_item_query>Slottet</search_item_query>
      </search_query>
   </search_request>

   <search_request transaction_id="IDSearch2">
      <search_request_header>
         <search_preferences>
            <search_settings matchtype="close"
                             wordmatch="beginning"
                             sorttype="confidence_sort" >
               <search_for_municipal/>
               <search_for_city/>
               <search_for_citypart/>
               <search_for_street/>
               <search_for_company/>
     <search_for_category/>
     <search_for_misc/>
          </search_settings>
        </search_preferences>
```

```
      </search_request_header>
      <search_query>
         <search_area_query>Göteborg</search_area_query>
         <search_item_query>station</search_item_query>
      </search_query>
   </search_request>

</isab-mc2>
```

48.6 Simple Route Request Example

An example of a route request to send to the $\mathcal{MC}^2$ 2-system.

```
<?xml version='1.0' encoding='iso-8859-1' ?>
<!DOCTYPE isab-mc2 SYSTEM 'isab-mc2.dtd'>

<isab-mc2>

   <auth>
      <auth_user>user</auth_user>
      <auth_passwd>pass</auth_passwd>
   </auth>

   <route_request transaction_id="IDRoute1">
      <route_request_header>
         <route_preferences route_description_type="normal">
            <route_settings route_vehicle="passengercar">
              <language>Swedish</language>
            </route_settings>
         </route_preferences>
      </route_request_header>

      <routeable_item_list>
         <position_item position_system="WGS84">
            <lat>N 552312</lat>
            <lon>E 131256</lon>
         </position_item>
      </routeable_item_list>

      <routeable_item_list>
         <position_item position_system="MC2">
            <lat>707789806</lat>
            <lon>215580176</lon>
         </position_item>
      </routeable_item_list>
   </route_request>

</isab-mc2>
```

48.7 Power Search

Power Search has a two round search procedure. The first round does a search within Wayfinders internal database, this is a fast search. The second round searches in external providers, this might take some time. So round 0 can be done first and display the results while round 1 search is beeing processed in the background for the clients.

The answer from Power Search is categoriesed into "headings". For example "Address", "POI", "YellowPages" etc. Each heading has an ID. This ID can be matched to a name and an image name in the search_desc_reply. The search_desc_reply is fetched using the search_desc_request with a language and a CRC. The CRC can either be empty or be a CRC from a previous search_desc_request. This CRC is used to reduce bandwidth.

If CRC matches the search_desc_reply then the reply will simply be crc_ok.

The search_desc_request is usually done by the client at start up.

Example Request:

```
<search_desc_request crc="" language="swe" transaction_id="id1"/>
```

Reply:

```
<search_desc_reply crc="2c499646" length="23" transaction_id="id1">
     <search_hit_type heading="0" round="0">
         <name>Platser</name>
         <image_name>search_heading_places</image_name>
      </search_hit_type>
      <search_hit_type heading="1" round="0">
         <name>Adresser</name>
         <image_name>search_heading_addresses</image_name>
      </search_hit_type>

      ... etc ...

</search_desc_reply>
```

To search with a position use the position_item node. Example:

```
<compact_search_request end_index="30"
                        language="swe" max_hits="30"
                        round="0" start_index="0"
                        transaction_id="id2" version="1">
    <search_item_query>Amst</search_item_query>
    <position_item position_system="MC2">
      <lat>626629220</lat>
      <lon>160005406</lon>
      <angle>0</angle>
    </position_item>
</compact_search_request>
```

and to fetch round 1 search do:

```
<compact_search_request end_index="30"
```

```
                          language="swe" max_hits="30"
                          round="1" start_index="0"
                          transaction_id="id2" version="1">
    <search_item_query>Amst</search_item_query>
    <position_item position_system="MC2">
      <lat>626629220</lat>
      <lon>160005406</lon>
      <angle>0</angle>
    </position_item>
 </compact_search_request>
```

Index