An interface between the Ministry of Transport's real time Data Center and applications developers
ICD specification for SIRI-SM protocol version 2.8

1. **General:**

   1.1. The Israeli Ministry of Transport (hereby: "MOT") manages a real-time data center for the public transportation (hereby: "Data Center"). The Data Center receives real-time information from the public transportation operator's systems, and sends the data to information boards at the stops, to public transportation application developers (hereby: "Developer"), and to the MOT's trip reporting system.

   1.2. This document (hereby: "ICD") focuses on the protocol between the Data Center and application Developers. The protocol set for this purpose is a standard protocol called SIRI-SM (hereby: "SM Protocol" or "SM").

   1.3. The European Standard Document CEN / EN 15531-1: 2012, also known as the "Service Interface for Real Time Information" (SIRI), is a complex technical specification that focuses on the information interfaces required for communication between technological systems in public transportation. The specification was created in cooperation with representatives of local authorities, operators, equipment suppliers and public representatives from eight European countries, with the support of the Standards Institute of Germany, France and the United Kingdom, on the basis of past standards defined for each country separately, by the European Standards Institute (CEN). The MOT documents will be based on document CEN / EN 15531: 2012 (hereby: "CEN") that is the official reference for the implementation of SIRI in this document.

   1.4. A CEN document consists of three volumes, with the relevant versions of this document in links:

   http://user47094.vs.easily.co.uk/siri/schema/2.0/docpdf/CEN_prEN_15531-1_(E)-SIRIv2.0_Part1-final_draft.pdf

   http://user47094.vs.easily.co.uk/siri/schema/2.0/docpdf/CEN_prEN_15531-2_(E)-SIRIv2.0_Part2-final_draft.pdf

   http://user47094.vs.easily.co.uk/siri/schema/2.0/docpdf/CEN_prEN_15531-3_(E)-SIRIv2.0_Part3-final_draft.pdf

1.5. Wherever a data is missing or there is apparent contradiction between the ICD document and the CEN documents, the ICD document prevail. It is clarified that defining fields as mandatory in the ICD document while the CEN is defined as an optional is possible. But the opposite definition is not supposed to exist in this document, <u>unless it is explicitly stated and explained.</u>

1.6. The SIRI specification covers several areas of activity, called "services", where each of the fields is defined by individual data structures that are relevant to it. The service that is relevant to this document is defined as SM, which are the initials of Stop Monitoring.

1.7. The choice of using a standard protocol allows, inter alia:

    1.5.1. Increasing collaboration and transparency between systems.

    1.5.2. Reducing the dependence on the way each system is implemented.

    1.5.3. Saving resources and preventing "dual work" in developing interfaces.

    1.5.4. Compliance with international systems and suppliers, and creating a competitive market.

1.8. The SM protocol transmits information about the expected arrival of a public transportation vehicle (herby: "Vehicle"), or a group of vehicles, to a stop ("תחנה"). For each Vehicle, the following information is transmitted, or in part:

    1.6.1. Trip details, according to the MOT licensing system ("מערכת הרישוי"), including: line ID, trip ID, vehicle number, origin stop number, destination stop number.

    1.6.2. Location on the global coordinate grid.

    1.6.3. Expected arrival time to the requested stop.

    1.6.4. Expected arrival times for next stops on route, if this data is requested (just at version 2.8).

1.9. The information transmitted through the SM protocol can be used by application developers for a number of purposes, including:

    1.7.1. Develop an application that show expected arrival of Vehicles to a stop.

    1.7.2. Trace the location of Vehicles, as part of an application.

    1.7.3. Use the data for research, or any other use.

1.8. Any interface developed on the basis of the ICD document can only use the services and fields defined in this document.

2. Configuration management and follow-up changes:

2.1. Revision history

Date	Edition	Description of the change
05/08/2012	2.6	First working version for application Developers.
15/11/2013	2.7	- Add location field. - Add trip ID field. - Enable compressed response.
30/04/2020	2.8 Sub-version 25	- Add SIRI-Lite request. - Add support for pre-defined filters that simplify the 'snapshot' request. - Add support for prediction to the stops after the requested monitored stop. - Add support for JSON response. - Remove 3 trips per line restriction.
15/05/2020	2.8 Sub-version 26	- Change the format of the response of 2.8 at sections 9.5, 9.6
17/6/2020	2.8 Sub-version 27	- Change the format of PreviewInternal at section 7.10
2/7/2020	2.8 Sub-version 28	- Change name of StopMonitoringDetailLevel to StopVisitDetailLevel at section 7.10
4/8/2020	2.8 Sub-version 29	- Fix Snapshot example at section 7.12
26/10/2020	2.8 Sub-version 30	- Add OperatorRef to Snapshots
15/2/2021	2.8 Sub-version 31	- Change links at sections 3.4

3. Compatibility

3.1. The Data Center supports the 2.7 and 2.8 versions only.

3.2. The MOT stops supporting the 2.7 ICD and this ICD describes both 2.7 and 2.8 versions. At any case of difference between these versions, the ICD will define it.

3.3. The MOT stops supporting version 2.6. The ICD does not refer to 2.6 at all. However a request with 2.6 version will be answered with 2.7 response.

3.4. Versions 2.7 and 2.8 uses different SIRI versions:

    3.4.1. Version 2.7 is based on SIRI 1.4, with the Schema files from: http://www.siri.org.uk/schema/1.4/siri-1.4.zip

    3.4.2. Version 2.8 is based on SIRI 2.0, with Schema files from: http://www.siri.org.uk/schema/2.0/Siri_XML-v2.0.zip

3.5. Versions 2.7 and 2.8 uses different request methods:

    3.5.1. Version 2.7 uses SOAP envelope and the request is at XML format.

    3.5.2. Version 2.8 uses SIRI-Lite and the request is defined by the URL parameters.

3.6. Versions 2.7 and 2.8 uses different response method:

    3.6.1. Version 2.7 uses XML response, wrapped with SOAP envelope.

    3.6.2. Version 2.8 uses XML or JSON response without SOAP envelope.

3.7. Version 2.8 adds 3 pre-defined 'snapshots' of all Vehicles, with different level of details.

3.8. Version 2.8 adds the support for prediction for stops after the requested monitored stop.

3.9. Beside the above changes, version 2.8 is backward compatible with version 2.7, so one can generate a 2.8 request that will have a response like 2.7.

3.10. **The MOT recommends any new developer to use version 2.8**.

3.11. In the future, the MOT plan to stop the support for 2.7 and support only 2.8 and newer versions.

4. Static data

4.1. Static data is all the data about a Vehicle trip, which is known before the trip started. This data include, inter alia, the route shape, the planned schedule, the stops sequence, the stops locations and there details.

4.2. The MOT uses the GTFS protocol (General Transit Feed Specification) for the static data, and the Developer should use both

the GTFS and the SM for a complete information about a trip. For example, the SM will use just the stop code number, and the stop name is defined in the GTFS.

4.3. More information about the GTFS can be found at: https://www.gov.il/he/departments/general/gtfs_general_transit_feed_specifications. The page is written in Hebrew, but can be translated with a supporting browser, like Chrome. The page contains a full detailed document, written in English.

4.4. The naming convention between the SM and the GTFS is very different. In order to match a fields, with same data, you can use the following table:

Field description	The name of the field in SIRI	The name of the field in GTFS
Line ID	LineRef	route_id
Operator	OperatorRef	agency_id
Direction	DirectionRef (valid values are 1,2,3)	direction_id (valid values are 0,1,2)
Line signage (" שילוט "קו")	PublishedLineName	route_short_name
Stop code (" מק"ט "תחנה")	DestinationRef, StopPointRef, OriginRef	stop_code
Stop order	Order	stop_sequence
Trip ID ("מזהה נסיעה")	FramedVehicleJourneyRef and its DataFrameRef, DatedVehicleJourneyRef sub fields	TripId at TripIdToDate.txt
Scheduled departure trip start	OriginAimedDepartureTime	departure_time at stops_times.txt

5. Request concept

5.1. The SM Request defines one stop, or more, that are the Monitored stop/s.

5.2. The SM protocol is mainly for delivering the details of expected arrivals of Vehicles to the Monitored stop/s.

5.3. The Request can include some filters that enable:

    5.3.1. Minimize the Response data, just to the exact data that is needed.

    5.3.2. For version 2.8, enhance the Response to enable 3 different 'Snapshot' mode, which gives a full pictures of all Vehicles in one Request.

5.4. The 2.8 and 2.7 versions have some differences in the Request, so the Request will be define in different sections for each version.

5.5. The definition of Request parameters is according the following rules:

    5.5.1. '1:1' indicates mandatory parameter, which must be included once in the Request, otherwise the Response will contain error message.

    5.5.2. '1:*' indicates mandatory parameter, which must be included one or more times in the Request, otherwise the Response will contain error message

    5.5.3. '0:1' indicates optional parameter, which can be included only once.

    5.5.4. '0:*' indicates optional parameter, which can be included one or more times.

6. Request for version 2.7

6.1. **The 2.7 is just for backward compatibility to existing Developers.**

6.2. **New Developers are strongly recommended NOT to use the 2.7 version, and use version 2.8 instead!!!!!!!!!!**

6.3. The Request is at SOAP, which is based on HTTP POST.

6.4. The Request address is fixed, and all the parameters are included in the HTTP POST body.

6.5. The Request address will be given to the Developer by Email, with his RequestorRef (the user name).

6.6. The Request is at XML format.

6.7. The HTTP request header should contain:

Accept: text/xml, multipart/related

Content-Type: text/xml; charset=utf-8

6.8. In order to minimize the size of the Response, and reduce the communication traffic, it is **strongly recommended** that the Response will be delivered compressed. In order to enable that, the HTTP request header should contain:

Accept-Encoding: gzip

6.9. The Request can be generated by a WDSL file, but it is **strongly recommended** to build the Request XML, and send it as HTTP POST, and by that:

6.9.1. The Developer can get the response compressed, and the communication link can be at a lower speed.

6.9.2. The Developer can control the 'connect' and 'read' timeout values.

6.9.3. The Developer can parse the Response with different ways, which are faster than the compiled WSDL.

6.10. At 2.7 the maximum arrivals to a stop, per LineRef, is limited to 3.

6.11. The Data Center will ignore MinimumStopVisitsPerLine in case it will be included in the Request.

6.12. The type of the fields should comply with the definitions at http://user47094.vs.easily.co.uk/siri/schema/1.4/siri-1.4.zip, for example, 'PositiveDuration' type is defined at siri_types-v1.1.xsd as

```
<xsd:simpleType name="DurationType">
  <xsd:restriction base="xsd:duration"/>
</xsd:simpleType>
<xsd:simpleType name="PositiveDurationType">
  <xsd:restriction base="DurationType"/>
</xsd:simpleType>
```

6.13. The Request XML is based on CEN Volume 3, Table 35, that was reduced at version 2.7 to the following table:

Hierarchy	Group	Field		Type	Description
1	SOAP	Envelope	1:1		SOAP Envelope
2	SOAP	Body	1:1		SOAP Body
3	SOAP	GetStopMonitoringService	1:1		SOAP service
4		Request	1:1		
5	*End Point Properties*	RequestTimestamp	1:1	*xsd:dateTime*	The time of the Request
		RequestorRef	1:1	*ParticipantCode*	User name that was given to the Developer by MOT.
		MessageIdentifier	0:1	*MessageQualifier*	Unique value for the current Request
6		StopMonitoringRequest	1:*		Start of a structure that defines a Monitored stop
	Atributes	Version	1:1		Attribute of StopMonitoringRequest. Should be set to '2.7'.

7		MessageIdentifier	0:1	*MessageQualifier*	Unique value for the current StopMonitoringRequest
7		PreviewInterval	0:1	*PositiveDuration*	Forward duration for which Stop Visits should be included, that is, interval before predicted arrival at the stop for which to include visits: only journeys which will arrive or depart within this time frame will be returned. Default value: 30 minutes Example: PreviewInterval of 60 minutes is defined as PT60M.
7		StartTime	0:1	*xsd:dateTime*	Initial start time for PreviewInterval. Default value: the current time, as defined by RequestTimestamp.
7	*Topic*	MonitoringRef	1:1	*MonitoringCode*	Stop code as the Monitored stop, or 'all' if a value is defined for LineRef.
7		LineRef	0:1	*LineCode*	Filter the Response for just lines that match the LineRef. Default value: the Response will contains all expected arrivals to the Monitored stop.
7		MaximumStopVisits	0:1	*xsd:positiveInteger*	The maximum number of arrival or departure visits to include in a given delivery. The first n Stop Visits within the look-ahead window are included. Only visits within the PreviewInterval are returned. Default value: the Response will contains all expected arrivals to the Monitored stop within the PreviewInterval.
7		MinimumStopVisitsPerLine	0:1	*xsd:positiveInteger*	This value is ignored by the Data Center

6.14. The following is an example for stop code 32902 Request XML:

```xml
<?xml version="1.0" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:siri="http://www.siri.org.uk/siri"
  xmlns:acsb="http://www.ifopt.org.uk/acsb"
  xmlns:datex2="http://datex2.eu/schema/1_0/1_0"
  xmlns:ifopt="http://www.ifopt.org.uk/ifopt"
  xmlns:siriWS="http://new.webservice.namespace">
<SOAP-ENV:Body>
  <siriWS:GetStopMonitoringService>
   <Request>
    <siri:RequestTimestamp>2019-11-24T18:00:45.237+03:00</siri:RequestTimestamp>
    <siri:RequestorRef>FILL_USER_NAME</siri:RequestorRef>
    <siri:MessageIdentifier>AC:20191124:180045:237</siri:MessageIdentifier>
    <siri:StopMonitoringRequest version="2.7">
     <siri:RequestTimestamp>2019-11-24T18:00:45.237+03:00</siri:RequestTimestamp>
     <siri:MessageIdentifier>0</siri:MessageIdentifier>
     <siri:PreviewInterval>PT120M</siri:PreviewInterval>
     <siri:StartTime>2019-11-24T18:00:45.237+02:00</siri:StartTime>
     <siri:MonitoringRef>32902</siri:MonitoringRef>
     <siri:LineRef>991</siri:LineRef>
```

```
        <siri:MaximumStopVisits>100</siri:MaximumStopVisits>
      </siri:StopMonitoringRequest>
    </Request>
  </siriWS:GetStopMonitoringService>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

7. Request for version 2.8

7.1. The Request is at SIRI-Lite, which is based on HTTP GET. See CEN Volume 2, section 12, for more information about SIRI-Lite.

7.2. The Request address for version 2.8 is based on a base address and parameters. The Request fields are sent as URL parameters.

7.3. The ICD will show the base address as <base address>.

7.4. The Request address will be given to the Developer by Email, with his RequestorRef (the user name).

7.5. Due to the use of SIRI-Lite, the Request structure does not comply with the SIRI XSD.

7.6. Some mandatory fields at SIRI are not included at the Request, like RequestTimestamp.

7.7. The Request URL is built with the following:

7.7.1. The requested service, which is stop monitoring.

7.7.2. The Response format, which can be 'xml' or 'json'.

7.7.3. The access key, which received from MOT, and is similar to RequestorRef at version 2.7.

7.7.4. The Request parameters.

7.8. The format of the Request URL is detailed in the following table:

Base		Version		Response format		Access key	Request parameters
<base address>	/	2.8	/	xml or json	?	Key=….	par1&par2…

7.9. The parameters are build according the following rules:

7.9.1. Define a value to a parameter, is by: parameter=value.
For example: LineRef=20.

7.9.2. In order to define more than one parameters, the parameters are separated with '&'.
For example: LineRef=5&MonitoringRef =32902.

7.9.3. The order of the parameters is not important.

7.9.4. Parameters can have more than one value, and in this case the values should be separated with ','.

For example: MonitoringRef=32901,32902.

7.9.5. Only one parameter can have more than one value, so:

MonitoringRef=32901,32902&LineRef=789 is valid.

MonitoringRef=32901&LineRef=567,5637 is valid.

MonitoringRef=32901,32902&LineRef=567,5637 is <u>NOT</u> valid.

7.10. The Request URL parameters are based on CEN Volume 3, Table

35, that was reduced at version 2.8 to the following table:

Field		Type	Description

PreviewInterval	0:1	*xsd:duration*	Forward duration for which Stop Visits should be included, that is, interval before predicted arrival at the stop for which to include visits: only journeys, which will arrive or depart within this time span will be returned.
			See - http://www.datypic.com/sc/xsd/t-xsd_duration.html
			Example:
			  PT45M = 45 minutes
			  PT1H = 1 hours
			Default value: 30 minutes.

StartTime	0:1	*See section 7.11*	Start time of the PreviewInterval time window. See section 7.11 for details about the format of this field. Default value: the current time.
MonitoringRef	1:1	*MonitoringCode*	The Monitored stop code, or pre-defined code. See section 7.14,7.13 for details about pre-defined values. See sections 7.9.4, 7.9.5 about defining more than one stop.
LineRef	0:1	*LineCode*	Filter the Response just for trips with the requested LineRef. Default value: show all lines. See sections 7.9.4, 7.9.5 about defining more than one line.
MaximumStopVisits	0:1	*xsd:positiveInteger*	Filter the Response to contain no more than MaximumStopVisits stop visits. Default: no limit.
MaximumStopVisitsPerLine	0:1	*xsd:positiveInteger*	Filter the Response to contain no more MaximumStopVisitsPerLine trips per each LineRef. Default value: no limit. This parameters is for compatibility with past implementation of version 2.7. Set a value of 3 in order to emulate behavior of version 2.7.
StopVisitDetailLevel	0:1	StopVisitDetailLevelEnum	Detailed level of the Response. At this version, the following valued are: - normal: the Response will not contain OnwardCalls. - calls: the Response will contain OnwardCalls. Default value: normal
MaximumNumberOfCallsOnwards	0:1	xsd:positiveInteger	Maximum number of OnwardCall entities per Vehicle journey. Default value: no limit

7.11. The format for StartTime is: YYYYMMDDTHHmmSSPSS

7.11.1. Y – year

7.11.2. M – month

7.11.3. D – day of month

7.11.4. T - separator, no meaning

7.11.5. H – hour, at 24 format

7.11.6. m – minutes

7.11.7. S – seconds

7.11.8. P – separator, no meaning

7.11.9. S – offset from GMT time

7.11.10.　　Example: 20181125T214953P02 is equal to the normal dateTime format of 2018-11-25T21:49:53+02:00

7.12. Examples

7.12.1. In the examples, a dummy key 'DM1234' will be used.

7.12.2. Basic Request for stop 32902 with XML response

<base address>/2.8/xml?Key=DM1234&MonitoringRef=32902

7.12.3. Basic Request for stop 32902 with JSON response

<base address>/2.8/json?Key=DM1234&MonitoringRef=32902

7.12.4. Request for stops 32901,32902

<base address>/2.8/xml?Key=DM1234&MonitoringRef=32901,32902

7.12.5. Request for stops 32901, and 32902 with filter to LineRef 5 for both stops.

<base address>/2.8/xml?Key=DM1234&MonitoringRef=32901,32902&LineRef=5

7.12.6. Basic Request for stop 32902, for the coming 45 minutes.

<base address>/2.8/xml?Key=DM1234&MonitoringRef=32902&PreviewInternal=PT45M

7.12.7. Request for stops 32901, to LineRef 5 and 6.

<base address>/2.8/xml?Key=DM1234&MonitoringRef=32901&LineRef=5,6

7.12.8. The following in **NOT a valid** Request, since it contradict section 7.9.5.

<base address>/2.8/xml?Key=DM1234&MonitoringRef=32902,32901&LineRef=5,6

7.12.9. Request for stops LineRef 5 and all stops of the line.

<base address>/2.8/xml?Key=DM1234&MonitoringRef=all&LineRef=5

7.12.10.　　Request for stop 32902, with details about arrival time of each Vehicle to all its remaining stops.

<base address>/2.8/xml?Key=DM1234&MonitoringRef=32902&StopVisitDetailLevel=calls

7.12.11.　　Request for stop 32902, with details about arrival time of each Vehicle to its next 2 stops

<base address>/2.8/xml?Key=DM1234&MonitoringRef=32902&StopVisitDetailLevel=calls&MaximumNumberOfCallsOnwards=2

7.12.12. Request for a 'snapshot' all active trips, with minimal data about each trip. See sections 7.15 for details.

<base address>/2.8/json?Key=DM1234&MonitoringRef=AllActiveTripsFilter&StopVisitDetailLevel=normal

7.12.13. Request for a 'snapshot' all active trips, with Onward data about each trip. See sections 7.16 for details.

<base address>/2.8/json?Key=DM1234&MonitoringRef=AllActiveTripsFilter&StopVisitDetailLevel=calls

7.12.14. Request for a 'snapshot' all planned trips. See sections 7.17 for details.

<base address>/2.8/json?Key=DM1234&MonitoringRef=AllPlannedTripsFilter&StopVisitDetailLevel=calls

7.13. In order to simplify the Request for information about many trips, version 2.8 contains pre-defined filters, which enable 'snapshot' mode. The pre-defined filter replace the stop code at MonitoringRef. The following sections details the 3 possible values: 'all', 'AllActiveTripsFilter', 'AllPlannedTripsFilter'.

7.14. MonitoringRef=all:

7.14.1. This filter must come with a value for LineRef.

7.14.2. The Response will contain the expected arrivals of all Vehicles, with the requested LineRef, to all stops of the line.

7.14.3. This filter valid for version 2.7 too.

7.14.4. The response will contains a series of MonitoredStopVisit entities that each contains a Vehicle visit to a stop, for a Vehicle.

7.14.5. Please note that the OnwardCalls (at section 10.8) mechanism at version 2.8 is more compact than the 'all' filter.

7.15. MonitoringRef=AllActiveTripsFilter with StopMonitoringDetailLevel=normal:

7.15.1. The Response will contains details about all the Vehicles that are in an Active trip.

7.15.2. An Active starts when the Vehicle is at the origin stop, till it reaches the destination stop.

7.15.3. The Response will contains details about the trip, the Vehicle position, but no prediction is contained.

7.15.4. The Data Center generate a Response for this filter every 15 seconds, and the Developer gets the latest, and most updated Response for this filter.

7.15.5. This filter resemble the 'vehicle positions' at the GTFS-Realtime, but at a different format.

7.15.6. This filter is a 'snapshot'. See more details about snapshot at section 7.18.

7.16. MonitoringRef=AllActiveTripsFilter with StopMonitoringDetailLevel=calls:

7.16.1. The Response will contains details about all the Vehicles that are in an Active trip.

7.16.2. An Active starts when the Vehicle is at the origin stop, till it reaches the destination stop.

7.16.3. The Response will contains details about the trip, the Vehicle position, with predictions to all stops that the Vehicle has not visited.

7.16.4. The Data Center generate a Response for this filter every 30 seconds, and the Developer gets the latest, and most updated Response for this filter

7.16.5. This filter resemble the 'trip updates' at the GTFS-Realtime, but at a different format.

7.16.6. This filter is a 'snapshot'. See more details about snapshot at section 7.18.

7.17. MonitoringRef=AllPlannedTripsFilter with StopMonitoringDetailLevel=normal:

7.17.1. The Response will contains details about all planned trips, which are scheduled trips that are not yet an Active trip.

7.17.2. The Response will contains details about the predictions to all stops that the Vehicle is planned to visit.

7.17.3. The Data Center generate a Response for this filter every 60 seconds, and the Developer gets the latest, and most updated Response for this filter

7.17.4. The Response will contains only trips that are scheduled to depart no more than 4 hours from the Request time.

7.17.5. This filter is a 'snapshot'. See more details about snapshot at section 7.18.

7.18. At Snapshot mode, the following apply:

7.18.1. The Response will contains a series of MonitoredStopVisit entities.

7.18.2. A Developer can send a Request with that filter at rate of 15 seconds or more between each Request.

7.18.3. When this filter is used, the Request should **NOT** contain parameters for: PreviewInterval, StartTime, LineRef, MaximumStopVisits, MaximumStopVisitsPerLine, MaximumNumberOfCallsOnwards.

7.18.4. The Request can be just for json format. XML format is not supported.

7.18.5. The following table summarize the fields that will be included in the Response.

Field	Snapshot according to section 7.15	Snapshot according to section 7.16	Snapshot according to section 7.17
RecordedAtTime	Yes	Yes	No
ItemIdentifier	No	No	No
MonitoringRef	No	No	No
LineRef	Yes	Yes	Yes
DirectionRef	No	No	No
FramedVehicleJourneyRef	Yes	Yes	Yes
PublishedLineName	No	No	No
OperatorRef	Yes	Yes	Yes
DestinationRef	No	No	No
OriginAimedDepartureTime	Yes	Yes	Yes
VehicleLocation	Yes	Yes	No
ConfidenceLevel	No	Yes	No
Bearing	Yes	Yes	No
Velocity	Yes	Yes	No
VehicleRef	Yes	Yes	Yes
MonitoredCall	Yes	Yes	No
MonitoredCall.StopPointRef	Yes	Yes	No
MonitoredCall.Order	Yes	Yes	No
MonitoredCall.AimedArrivalTime	No	No	No
MonitoredCall.ExpectedArrivalTime	No	No	No
MonitoredCall.ArrivalStatus	No	No	No
MonitoredCall.ArrivalPlatformName	No	No	No
MonitoredCall.DistanceFromStop	Yes	Yes	No
OnwardCalls	No	Yes	Yes
OnwardCalls.StopPointRef	No	Yes	Yes
OnwardCalls.Order	No	Yes	Yes
OnwardCalls.ExpectedArrivalTime	No	Yes	Yes
OnwardCalls.ArrivalStatus	No	Optional	Optional
OnwardCalls.ArrivalPlatformName	No	Optional	Optional

8. Response at version 2.7

8.1. The Response at version 2.7 is XML wrapped by SOAP envelope.

8.2. The 2.7 and 2.8 have different wrapper, and the similarity is at the level of StopMonitoringDelivery and its sub fields. So the ICD will define the common fields at section 10, and this section will deal just with the fields outside StopMonitoringDelivery.

8.3. The following table describe the structure of the Response.

Hierarchy	Group	Field		Type	Description
1	SOAP	Envelope	1:1		SOAP Envelope
2	SOAP	Body	1:1		SOAP Body
3	SOAP	GetStopMonitoringServiceResponse	1:1		SOAP response
4		Answer	1:1		
5	*End Point Properties*	ResponseTimestamp	1:1	*xsd:dateTime*	The time of the Response
		ProducerRef	1:1	*ParticipantCode*	The version of the SM Server
		ResponseMessageIdentifier	1:1	*MessageQualifier*	An arbitrary unique reference associated with the response which may be used to reference it.
		RequestMessageRef	0:1	*MessageQualifier*	Reference to a unique message identifier associated with the request which gave rise to this response. The SM Server will put the value of MessageIdentifier from the Request.
		Status	1:1	*xsd:boolean*	Whether the complete request could be processed successfully or not.
		ErrorCondition	0:1	*See below*	
6		OtherError	1:1		
7		ErrorText	1:1		
6		Description	1:1	*ErrorDescription*	Description of Error.
6		StopMonitoringRequest	0:*		Start of a structure that defines a Monitored stop, see section 10 for details. In case of an error in the Request, this structure will not be included.

8.4. A basic Response may look like:

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
 <S:Body>
  <ns7:GetStopMonitoringServiceResponse xmlns:ns3="http://www.siri.org.uk/siri" xmlns:ns4="http://www.ifopt.org.uk/acsb" xmlns:ns5="http://www.ifopt.org.uk/ifopt"
xmlns:ns6="http://datex2.eu/schema/1_0/1_0" xmlns:ns7="http://new.webservice.namespace">
   <Answer>
    <ns3:ResponseTimestamp>2019-05-11T12:40:46.148+03:00</ns3:ResponseTimestamp>
    <ns3:ProducerRef>ISR Siri Server (141.13)</ns3:ProducerRef>
    <ns3:ResponseMessageIdentifier>8757681</ns3:ResponseMessageIdentifier>
    <ns3:RequestMessageRef>AC:20190511:124112:262</ns3:RequestMessageRef>
    <ns3:Status>true</ns3:Status>
    <ns3:StopMonitoringDelivery version="2.7">
     <ns3:ResponseTimestamp>2019-05-11T12:40:46.148+03:00</ns3:ResponseTimestamp>
     <ns3:Status>true</ns3:Status>
     <ns3:MonitoredStopVisit>
      <ns3:RecordedAtTime>2019-05-11T12:40:47.000+03:00</ns3:RecordedAtTime>
      <ns3:ItemIdentifier>503635267</ns3:ItemIdentifier>
      <ns3:MonitoringRef>51202</ns3:MonitoringRef>
      <ns3:MonitoredVehicleJourney>
       <ns3:LineRef>1209</ns3:LineRef>
       <ns3:DirectionRef>2</ns3:DirectionRef>
       <ns3:FramedVehicleJourneyRef>
        <ns3:DataFrameRef>2019-05-11</ns3:DataFrameRef>
        <ns3:DatedVehicleJourneyRef>20925867</ns3:DatedVehicleJourneyRef>
       </ns3:FramedVehicleJourneyRef>
       <ns3:PublishedLineName>1</ns3:PublishedLineName>
       <ns3:OperatorRef>6</ns3:OperatorRef>
       <ns3:DestinationRef>51202</ns3:DestinationRef>
       <ns3:OriginAimedDepartureTime>2019-05-11T12:00:00.000+03:00</ns3:OriginAimedDepartureTime>
       <ns3:VehicleLocation>
        <ns3:Longitude>35.3139762878418</ns3:Longitude>
        <ns3:Latitude>32.70824432373047</ns3:Latitude>
       </ns3:VehicleLocation>
       <ns3:VehicleRef>9030930</ns3:VehicleRef>
       <ns3:MonitoredCall>
```

```
        <ns3:StopPointRef>51202</ns3:StopPointRef>
        <ns3:ExpectedArrivalTime>2019-05-11T12:51:00.000+03:00</ns3:ExpectedArrivalTime>
      </ns3:MonitoredCall>
    </ns3:MonitoredVehicleJourney>
   </ns3:MonitoredStopVisit>
  </ns3:StopMonitoringDelivery>
  </Answer>
 </ns7:GetStopMonitoringServiceResponse>
 </S:Body>
</S:Envelope>
```

8.5. A Response with an error may look like:

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
 <S:Body>
  <ns7:GetStopMonitoringServiceResponse xmlns:ns3="http://www.siri.org.uk/siri"
   xmlns:ns4="http://www.ifopt.org.uk/acsb" xmlns:ns5="http://www.ifopt.org.uk/ifopt"
   xmlns:ns6="http://datex2.eu/schema/1_0/1_0" xmlns:ns7="http://new.webservice.namespace">
   <Answer>
    <ns3:ResponseTimestamp>2019-05-13T11:54:45.482+03:00</ns3:ResponseTimestamp>
    <ns3:ProducerRef>ISR Siri Server (141.10)</ns3:ProducerRef>
    <ns3:ResponseMessageIdentifier>45032656</ns3:ResponseMessageIdentifier>
    <ns3:RequestMessageRef>AC:20170425:230410:981</ns3:RequestMessageRef>
    <ns3:Status>false</ns3:Status>
    <ns3:ErrorCondition>
     <ns3:OtherError>
      <ns3:ErrorText>720</ns3:ErrorText>
     </ns3:OtherError>
     <ns3:Description>User authentication failed for FIIL_YOUR_USERNAME from 31.168.45.66</ns3:Description>
    </ns3:ErrorCondition>
   </Answer>
  </ns7:GetStopMonitoringServiceResponse>
 </S:Body>
</S:Envelope>
```

9. Response at version 2.8

9.1. The Response at version 2.8 can be XML or JSON according to the Request URL.

9.2. The XML format of the Response should be valid SIRI 2.0, as checked against siri.xsd at http://user47094.vs.easily.co.uk/siri/schema/2.0/Siri_XML-v2.0.zip

9.3. The Developer can parse the XML with the above siri.xsd, or with any other way to parse XMLs.

9.4. The following table describe the structure of the Response:

Hierarchy	Group	Field		Type	Description
1		Siri	1:1		SOAP response
2		ServiceDelivery	1:1		
3	*End Point Properties*	ResponseTimestamp	1:1	*xsd:dateTime*	The time of the Response
		ProducerRef	1:1	*ParticipantCode*	The version of the SM Server
		ResponseMessageIdentifier	1:1	*MessageQualifier*	An arbitrary unique reference associated with the response which may be used to reference it.
4		StopMonitoringRequest	1:*		Start of a structure that defines a Monitored stop, see section 10 for details. In case of an error in the Request, this structure will contain the error.

9.5. A basic Response may look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<Siri xmlns="http://www.siri.org.uk/siri">
 <ServiceDelivery>
  <ResponseTimestamp>2019-05-11T12:40:46.148+03:00</ResponseTimestamp>
  <ProducerRef>MOT_Data_Center_V154_3</ProducerRef>
  <StopMonitoringDelivery version="2.8">
   <ResponseTimestamp>2019-05-11T12:40:46.148+03:00</ResponseTimestamp>
   <ValidUntil>2019-05-11T22:40:46.148+03:00</ValidUntil>
```

```
<MonitoredStopVisit>
  <RecordedAtTime>2019-05-12T15:00:42-04:00</RecordedAtTime>
  <ItemIdentifier>503635267</ItemIdentifier>
  <MonitoringRef>51202</MonitoringRef>
  <MonitoredVehicleJourney>
    <LineRef>1209</LineRef>
    <DirectionRef>2</DirectionRef>
    <FramedVehicleJourneyRef>
      <DataFrameRef>2019-05-11</DataFrameRef>
      <DatedVehicleJourneyRef>20925867</DatedVehicleJourneyRef>
    </FramedVehicleJourneyRef>
    <PublishedLineName>1</PublishedLineName>
    <OperatorRef>6</OperatorRef>
    <OriginRef>51102</OriginRef>
    <DestinationRef>51202</DestinationRef>
    <ConfidenceLevel>probablyReliable</ConfidenceLevel>
    <VehicleLocation>
      <Longitude>-73.991028</Longitude>
      <Latitude>40.663611</Latitude>
    </VehicleLocation>
    <Bearing>43.40886</Bearing>
    <VehicleRef>9030930</VehicleRef>
    <MonitoredCall>
      <StopPointRef>52202</StopPointRef>
      <Order>5</Order>
      <ExpectedArrivalTime>2019-05-12T15:12:02.569-04:00</ExpectedArrivalTime>
    </MonitoredCall>
  </MonitoredVehicleJourney>
</MonitoredStopVisit>
      </StopMonitoringDelivery>
   </ServiceDelivery>
</Siri>
```

9.6. A Response with and error may look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<Siri xmlns="http://www.siri.org.uk/siri">
  <ServiceDelivery>
    <ResponseTimestamp>2019-05-13T05:32:40.828-04:00</ResponseTimestamp>
    <StopMonitoringDelivery version="2.8">
      <ResponseTimestamp>2019-05-13T05:32:40.828-04:00</ResponseTimestamp>
      <ErrorCondition>
        <OtherError>
          <ErrorText>API key is not authorized</ErrorText>
        </OtherError>
      </ErrorCondition>
    </StopMonitoringDelivery>
  </ServiceDelivery>
</Siri>
```

9.7. The following are possible error massages:

9.7.1. "API key is not authorized".

9.7.2. "No such route: 3415".

9.7.3. "No such stop: 4566".

10. Common information for Response at both versions 2.7 and 2.8

10.1. As mentioned before, the versions 2.7 and 2.8 has the StopMonitoringDelivery structure in common, and are different outside this structure.

10.2. This section will detail the fields at StopMonitoringDelivery.

10.3. The hierarchy column will be relative to the hierarchy of StopMonitoringDelivery. The symbol 'S' will resemble the hierarchy of StopMonitoringDelivery, so 'S+1' will be a field at one level inside StopMonitoringDelivery.

10.4. The following table details the fields that are inside StopMonitoringDelivery. The table is based on CEN Volume 3 table 41, and CEN Volume 2 table 16:

Hierarchy	Versions	Group	Field		Type	Descriptions

S	both	Attributes	version	1:1	*VersionString*	Fix to '2.7' or '2.8'. For version, the Response will return the version in the Request and values like '2.7IL' are also valid.
S+1	both	Log	ResponseTimestamp	1:1	*xsd:dateTime*	The time of the Response.
S+1	both		Status	1:1	xsd:boolean	Whether the complete request could be processed successfully or not.
S+1	both		ErrorCondition	1:0		
S+2	both	Status	OtherError	1:1		
S+3	both		ErrorText	1:1	*xsd:string*	At 2.7 the field contains the error code. At 2.8 the field contains the error description.
S+2	2.7		Description	1:1	*xsd:string*	The description of the error
S+1	both	Payload	MonitoredStopVisit	0:*	*See section 0*	A visit to a Monitored stop by a Vehicle as an arrival and /or departure.

10.5. The following table details the fields that are inside

MonitoredStopVisit. The table is based on CEN Volume 3 table 41:

Hierarchy	Versions	Group	Field		Type	Descriptions
S+2	both	Log	RecordedAtTime	1:1	*xsd:dateTime*	Time at which data was recorded at the Vehicle. That mean, for example, that the Vehicle was at the VehicleLocation at RecordedAtTime.
S+2	both	Identity	ItemIdentifier	0:1	*ItemIdentifier*	Unique identifier of the MonitoredStopVisit.
S+2	both	StopVisitReference	MonitoringRef	1:1	*MonitoringCode*	The monitored stop code.
S+2	both	JourneyInfo	MonitoredVehicleJourney	1:1	See section 10.6	

10.6. The following table details the fields that are inside

MonitoredVehicleJourney. The table is based on CEN Volume 3 table 43.

Hierarchy	Versions	Group	Field		Type	Descriptions
S+3	both	VehicleJourneyIdentity	LineRef	1:1	*LineCode*	Reference to a LINE
S+3	both		DirectionRef	1:1	*DirectionCode*	Reference to a DIRECTION the VEHICLE is running along the LINE. Valid values are: 1, 2, 3.

S+3	both		FramedVehicle JourneyRef	1:1	*FramedVehicleJourneyRefStructure*	Trip ID structure. See also CEN Volume 2 table 63.
S+4	both	FramedVehicleJourneyRef	DataFrame Ref	1:1	*DataFrameQualifier*	The date part of the trip ID. Valid value is like: 2019-05-11.
S+4	both		DatedVehicleJourneyRef	1:1	*DatedVehicleJourneyCode*	The number part of trip ID. Valid value is like: 20925867. The value is reference to TripId at TripIdToDate.txt file at the GTFS.
S+3	both	JourneyPatternInfoGroup	PublishedLineName	1:1	*NLString*	The bus number, as published on the bus. Valid values are like: 561, 99א,7 The value is reference to route_short_name at the GTFS. See also CEN Volume 2 table 70.
S+3	both	ServiceInfoGroup	OperatorRef	1:1	*OperatorCode*	The Operator code. The value is reference to agency_id at the GTFS. See also CEN Volume 2 table 67.
S+3	both	VehicleJourneyInfoGroup	DestinationRef	1:1	*JourneyPlaceCode*	The destination stop code . The value is reference to stop_code at the GTFS. See also CEN Volume 2 table 68.
S+3	both		OriginAimedDepartureTime	1:1	*xsd:dateTime*	The start time of the Journey, according to the licensing system ("מערכת הרישוי"). The value should match DepartureTime at TripIdToDate.txt file at the GTFS, for the relevant TripId.
S+3	2.8	Journey ProgressInfo	ConfidenceLevel	0:1	*certain | veryReliable | reliable | probablyReliable | unconfirmed*	The confidence level of the prediction.
S+3	both		VehicleLocation	0:1		See also CEN Volume 2 table 64.
S+4	both		Longitude	1:1	*LongitudeType*	Latitude from equator. Valid value may be like: 35.3139762.
S+4	both		Latitude	1:1	*LatitudeType*	Latitude from equator. Valid value may be like:
S+3	2.8		Bearing	0:1		Vehicle bearing with respect to the North. Valid values are between 0 and 360.
S+3	2.8		Velocity	0:1		Vehicle speed at Km/h.
S+3	both	OperationalInfoGroup	VehicleRef	1:1	*VehicleCode*	Vehicle number. The value should match the license number of the Vehicle ("מספר רישוי"). Valid value may be like: 9030930
S+3	both	CallingPattern	MonitoredCall	0:1	Prediction for the monitored stop. See section 10.7	
S+3	2.8		OnwardCalls	0:1		
S+4	2.8		OnwardCall	1:*	See section 10.8	

10.7. MonitoredCall fields.

    10.7.1. The MonitoredCall fields have 2 different meaning, according to the Request.

    10.7.2. Mode A – normal stop monitoring Request, about Vehicle arrivals to Monitored stop.

    10.7.3. Mode B – Valid for Requests at 'snapshot' mode according to sections 7.15, 7.16, 7.17. Also for Requests with StopMonitoringDetailLevel=calls.

    10.7.4. The following table details the fields that are inside MonitoredCall. The table is based on CEN Volume 3 table 45.

Hierarchy	Versions	Group	Field		Type	Description at mode A	Description at mode B
S+4	both	StopIdentity	StopPointRef	1:1	*MonitoringCode*	Monitored stop code.	The stop code of the stop that the Vehicle is stopping at now, or recently visited.
S+4	2.8		Order	0:1	*xsd:positiveInteger*	Monitored stop order. The first stop is at Order 1.	The stop order of the stop that the Vehicle is stopping at now, or recently visited
S+4	both	Arrival	AimedArrivalTime	0:1	*xsd:dateTime*	Planned arrival time to the Monitored stop. <u>The value will be pass only when the Vehicle have **not** started the journey.</u> The value, if any, will be the same as ExpectedArrivalTime.	Will not pass
S+4	both		ExpectedArrivalTime	0:1	*xsd:dateTime*	<u>Expected arrival time to the Monitored stop.</u> <u>The value will be pass both when the Vehicle have started or not the journey</u>.	Will not pass
S+4	2.8		ArrivalStatus	0:1	*onTime | early | delayed | cancelled | arrived | noReport*	Arrival status to the Monitored stop.	Will not pass
S+4	2.8		ArrivalPlatformName	0:1	*NLString*	Arrival platform name, mainly for trains.	Will not pass
S+4	2.8		DistanceFromStop	0:1	*DistanceType*	The distance that the Vehicle travelled from the start of the journey. Note: MOT defines this field in a different way that CEN document.	

							The field value will have the same value of LinkDistance at the SIRI-VM from the Operators. The distance will be in units of Meters.

10.8. The following table details the fields that are inside OnwardCall.

10.9. The OnwardCall entities will contain prediction for all the stops, starting from the next stop that the Vehicle is schedule to arrive, till the last stop of the line.

10.10. Example: Assume that a line has 10 stops. The Vehicle is between stop 4 and 5. So there will be OnwardCall entities for stop 5,6,7,8,9,10.

10.11. The Onward calls are independent of the MonitoredCall, and there may be duplication between an OnwardCall entity and MonitoredCall.

10.12. The mechanism for selecting stops to be shown as Onwardcall is similar to the [MTA SIRI-SM](#).

10.13. The table is based on CEN Volume 3 table 48.

Hierarchy	Versions	Group	Field		Type	Descriptions
S+5	2.8		StopPointRef	1:1	MonitoringCode	Stop code
S+5	2.8		Order	1:1	xsd:positiveInteger	Stop order. The first stop is with Order 1
S+5	2.8		ExpectedArrivalTime	11:	xsd:dateTime	Expected arrival time
S+5	2.8	StopId entity	ArrivalStatus	0:1	onTime | early | delayed | cancelled | arrived | noReport	Arrival status to the stop. If the arrival to the stop is cancelled, at value of 'cancelled' will be passed.
S+5	2.8		ArrivalPlatformName	0:1	NLString	Arrival platform name, mainly for trains.

11. SIRI General Capabilities

11.1. When implementing SIRI-SM protocol, based on CEN documents, several decision must be taken. This chapter, and the following chapter, details the decisions that MOT took for the Israeli SIRI-SM version 2.7 and 2.8.

11.2. The decisions are detailed by a mechanism that is called "Capability Matrix", and is described at CEN Volume 2, table 6.

11.3. The following table is the MOT decisions based on CEN Volume 2, table 6.

Subject	Definition		
Setting Up Services	SM		
Defines the version of SIRI on which it is based	The document hierarchy is : A. The tables of this document (ICD). B. Scheme files. C. CEN documents.		
Determination of communication method	Request-Response		
The protocol	2.7: SOAP, that is based on HTTP POST 2.8: SIRI-Lite, that is based on HTTP GET		
Ability Matrix	**Subject**	**Decision**	**Implication**
All mandatory fields in the general capabilities matrix should be exercised according to CEN Volume 2, Table 6. In addition, the optional fields must be used in the manner shown in the table on the left. There is no need to pass the capacity matrix in any message since the determinations in this regard are permanent.	*Versioning*	YES	Check version in each request.
	CapabilityChecking	NO	Checking the capabilities of the server to provide what is required of the client request. If not, a transmission error is returned - CapabilityNotSupportedError
	CapabilityDiscovery	NO	What is determined in the ICD will be exercised statically
	DynamicContext	NO	Same as above.
	Historic	NO	
	MultipleFilters	NO	
	AccessControl	YES	Permission option.
	DirectDelivery	YES	Forward the entire response directly after the request without intermediate messages.
	VisitCountIsOrder	NO	
	MultipartDespatch	NO	Option to split messages.

Subject	Definition			
	CheckStatus	NO	Ability to check server status by the client.	
	Heartbeat	NO	Using the check-status instead.	
	SoapEnvelope	2:7 YES 2.8 NO		
	Addresses	Implicit	Static server addresses settings.	
	Compression	YES	Support the option of gzip compression to the response.	
Is IP Authentication or another method?	IP Authentication. Call from an unknown address will result in an error.			
Access rights	On the basis of the AccessControl definitions.			

12. Basic settings for the interface ("capacity matrix" for the SM service):

12.1. The purpose of this section and its table is to summarize the server capabilities and interface.

12.2. The section is based on the Capability Matrix of the SM service, as expressed in CEN Volume 3, Table 33, with local adjustments.

12.3. Rules:

12.3.1. The table below defines the frame boundaries for the tables in the following sections.

12.3.2. The table is static and cannot be dynamically changed while running.

12.3.3. The capability defined in this table as False will not be supported by the SM Server.

12.3.4. The capability defined in this table as True will be supported by the SM Server, but the SM client does not have to use the specific capability.

12.3.5. The SM client is responsible for transferring only requests that meet the definitions in the table.

Topic	Capability	Value	Description
TopicFiltering	**Which optional filtering features are supported?**		
	DefaultPreviewInterval	30 minutes	Default Preview Interval.
	ByStartTime	True	Whether a start time other than now can be specified for preview interval.
	FilterByMonitoringRef	True	Whether results can be filtered by Monitoring point, which is a stop number or a pre-defined of a group of stops.

	FilterByLineRef	True	Whether results can be filtered by **LineRef**.
	FilterByDirectionRef	False	Whether results can be filtered by **DirectionRef**.
	FilterByDestination	False	Whether results can be filtered by **DestinationRef**.
	FilterByVisitType	False	Whether results can be filtered by **VisitType**, e.g. arrivals, departures
RequestPolicy	**Which features of RequestPolicy are supported by service?**		
	Language	English	Languages used by service.
	Translations	False	Whether the producer supports translations.
	GmlCoordinateFormat	WGS84	Coordinate format.
	WgsDecimalDegrees		The coordinate will be delivered as decimal numbers like 32.54672,34.67853
	UseReferences	True	Whether results can return references for stops.
	UseNames	False	Whether results can return names for stop.
	HasDetailLevel	2.7: False 2:8 True	Whether Detail level filtering is supported.
	DefaultDetailLevel	2.7 N/A 2.8 Normal	Default Detail level if none specified on request.
	HasMaximumVisits	True	Whether results can be limited to a maximum number of arrival visits to include in a given delivery.
	HasMinimumStopVisitsPerLine	False	Whether results can be limited to include a minimum number of arrivals visits for a given LineRef to include in a given delivery.
	HasMaximumStopVisitsPerLine	2.7 fixed to a value of 3 2.8 True	Whether results can be limited to include a maximum number of arrivals visits for a given LineRef to include in a given delivery. **This is not a not part of SIRI, and is added to version 2.8 just to enable backward compatibility with past implementation of version 2.7.** <u>Version 2.7 has a fixed maximum of 3 arrivals visits for a given LineRef to a stop.</u>
	HasNumberOfOnwardsCalls	2.7 False 2.8 True	If system can return detailed calling pattern, whether a number of ONWARDS CALLS to include can be specified?
	HasNumberOfPreviousCalls	False	If system can return detailed calling pattern, whether a number of PREVIOUS CALLS to include can be specified?
Subscription Policy	Not relevant since Request Response mode is defined.		
AccessControl	**Which optional Access Control features are supported by service?**		
	RequestChecking	False	Whether Access Control of requests is supported?
	CheckOperatorRef	False	If access control is supported, whether access control by OPERATOR is supported.

	CheckLineRef	False	If access control is supported, whether access control by LINE is supported.
	CheckMonitoringRef	False	If access control is supported, whether access control by monitoring point (LOGICAL DISPLAY) is supported.
ResponseFeatures			
	HasLineNotice	False	Whether service supports Line Notices?