ACT Government

NXTBUS

Developer's API Reference

Version: v1.12

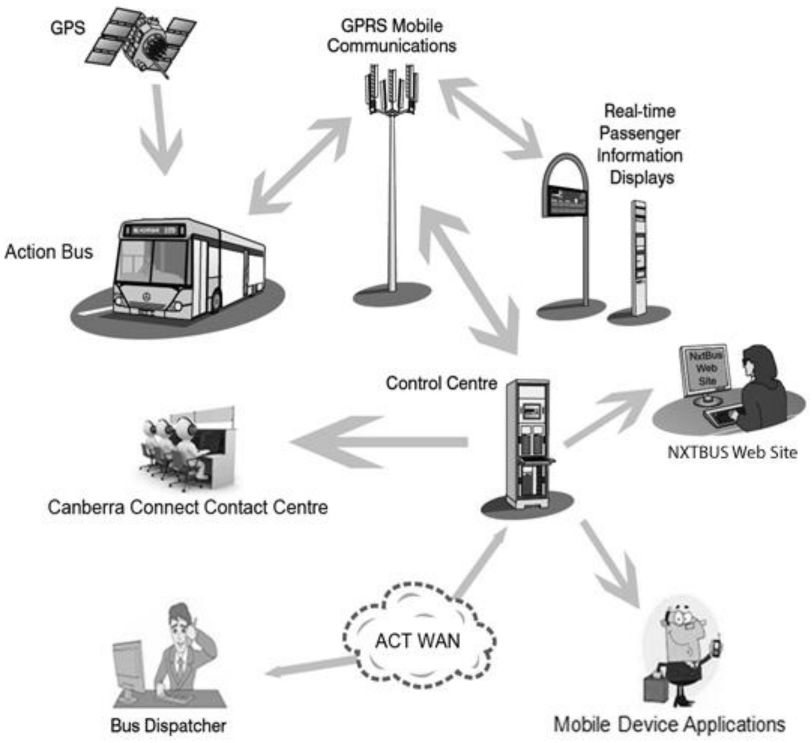
Contents

1 Introduction

NXTBUS is the real time passenger information system used to track ACTION buses and provide information on bus arrival and departure times. The implementation of the system was an initiative of the ACT Government's Territory and Municipal Services Directorate (TAMS).

NXTBUS captures bus location data in real-time from GPS tracking equipment that has been installed in ACTION buses. Bus location information is transmitted over GSM/GPRS networks through the central NXTBUS system and made available to commuters via:

- Digital displays fitted at bus stations and on busses
- Information kiosks located at bus terminals
- Mobile phones using SMS messaging
- Personal computers, smart phones, and tablets via the NXTBUS website <u>or third-party developed applications</u>


GPS
GPRS Mobile Communications
Real-time Passenger Information Displays
Action Bus
Control Centre
NXTBUS Web Site
Canberra Connect Contact Centre
ACT WAN
Bus Dispatcher
Mobile Device Applications

The NXTBUS system incorporates an Application Programming Interface (API) that can be used by third-party developed software to access real time transit information about the ACTION bus network, including bus arrival and departure times. The NXTBUS API is based on the Service Interface for Real Time Information (SIRI) protocol.

SIRI is an XML protocol that allows distributed computers to exchange real time information about public transport services and vehicles. The protocol is a Comité Européen de Normalisation (CEN) standard that is based on the Transmodel Reference Data Model for public transport information.

1.1 Purpose

This document aims to inform those seeking to develop applications that interface with NXTBUS. It is expected that parties seeking to integrate with the NXTBUS interface will take the requisite care in researching and understanding the information provided in this document and the supporting Reference documentation.

1.2 General Contextual Information

The SIRI Standard, as defined in the SIRI standards documentation (References ([1], [2] & [3]), incorporates a comprehensive set of functions that deliver a variety of data feeds. The implementation of the NXTBUS API currently supports the following SIRI data feeds:

- Production Timetable data,
- Estimated Timetable data,
- Bus Stop Monitoring, and
- Vehicle Monitoring.

The element descriptions for the API functions related to these services are the only ones that have been implemented by NXTBUS. Data elements and messages not included in this document are not supported.

There are factors (both internally and externally) that can affect accuracy of the real-time location data, the impacts of which includes:

- a maximum time-delay of position data of approximately 60 seconds,
- the maximum error for distance information is 450m, and
- the maximum error for GPS co-ordinates is 900m.

1.3 Obtaining an API Key

As supplied from the initial registration to the NXTBUS interface. The value of the API key supplied, then serves as the value for instances of Requestor code (RequestorRef & SubscriberRef).

1.4 Obtaining Lookup and Reference Data

Most services offered by NXTBUS require the consumer to provide a reference code as an input parameter. This section describes the format and values accepted by these elements.

1.4.1 LineRef

LineRef element follows the format "ACT_NNN", where

- NNN = three digit number of the bus route, e.g. "200"

For example, the LineRef to use when looking for information on bus route 200 (Gungahlin-Fyshwick) would be = “ACT_200”

1.4.2 DirectionRef

DirectionRef element follows the format “ACT_NNN_D”, where

- NNN = three digit number of the bus route, e.g. “200”
- D = direction of the trip, e.g. “A” or “B”

For example, the DirectionRef to use when looking for information on bus route 200 (Gungahlin-Fyshwick) would be = “ACT_200_A” (to Gungahlin) and “ACT_200_B” (to Fyshwick).

1.4.3 VehicleMonitoringRef (for Vehicle Monitoring)

VehicleMonitoringRef element follows the format “VM_ACT_LLLL”, where

- LLLL = four digit number of the bus route, e.g. “0200”

For example, the VehicleMonitoringRef to use when looking for information on the vehicles travelling on bus route 200 (Gungahlin-Fyshwick) would be = “VM_ACT_0200”.

1.4.4 MonitoringRef (for Stop Monitoring)

When used in StopMonitoring service calls, this element follows the same format as the StopPointRef element.

1.4.5 StopPointRef

StopPointRef element follows the format “SSSS”, where

- SSSS = four digit number of the stop point, e.g. “2245”

The latest list of stop point ids can be acquired from http://www.action.act.gov.au/rider_Info/apps.

1.5 Terms of Use

All SIRI data is provided under a Creative Commons Attribution license, as detailed at the following location: http://creativecommons.org/licenses/by/3.0/au/deed.en.

Additionally, the SIRI Standard itself has some inherent Terms of Use, as described at the following location: http://user47094.vs.easily.co.uk/siri/termsOfUse.htm.

1.6 Glossary

Term	Description
ACTION	ACT Internal Omnibus Network
API	Application Programming Interface
ET	Estimated Timetable
HTTP	Hyper Text Transfer Protocol
NXTBUS	Real Time Passenger Information System for the ACTION network
PT	Production Timetable
R&PTD	ACT Roads and Public Transport Division
SIRI	Service Interface for Real Time Information
SM	Stop Monitoring
TAMS	Territory and Municipal Services
URL	Universal Resource Locator
VM	Vehicle Monitoring
XML	Extensible Markup Language

1.7 Additional Reference Material

[1]	CEN/TS 00278181-1 Part 1: Introduction	#183645
[2]	CEN/TS 00278181-2 Part 2: Communications infrastructure	#183646
[3]	CEN/TS 00278181-3 Part 3: Functional Service Interfaces	#183647
[4]	SIRI Web Site: www.siri.org.uk	
[5]	Trapeze SIRI Interface Specification(How-To) for TAMS NXTBUS	

2 Common communication aspects

2.1 SIRI-Standard

The NXTBUS SIRI Interface was implemented in accordance with the draft SIRI v2.0 Standard

2.2 Data Exchange Patterns of Interaction

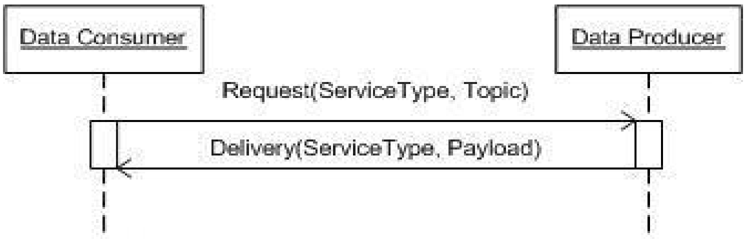
2.2.1 Synchronous Responses

All NXTBUS requests are responded to synchronously.

Although NXTBUS supports asynchronous data fetching through its publish/subscribe protocols, **every request message will receive a synchronous response message**. In some cases the response message will contain just an acknowledgement of the request.

2.2.2 Request/Response

The Request/Response interaction allows for the immediate fulfilment of one-off data supply requests made by a Requestor to a Service



The requestor sends a Service Request to a Specific SIRI Functional Service as located by the Service’s Endpoint Reference, and is returned an immediate data delivery, i.e. a response message that contains application payload data.

Each specific SIRI Functional Service Request is wrapped within a general ServiceRequest element, and the corresponding delivery is similarly wrapped within a ServiceDelivery element.

NXTBUS currently supports the Request/Response protocol for Stop Monitoring, Vehicle Monitoring and Production Timetable.
Note that while **EstimatedTimetable request/response messaging** is documented in the Trapeze SIRI Interface Specification it is **not currently supported**.

HTTP-request	HTTP-response
ProductionTimetableRequest	ProductionTimetableResponse
StopMonitoringRequest	StopMonitoringResponse

VehicleMonitoringRequest	VehicleMonitoringResponse

2.2.3 Publish/Subscribe

The following is a summary of the supported http request/response pairs:

HTTP-request	HTTP-response
CheckStatusRequest	CheckStatusResponse
SubscriptionRequest	SubscriptionResponse
TerminateSubscriptionRequest	TerminateSubscriptionResponse
DataReadyNotification	DataReadyAcknowledgement
DataSupplyRequest	ServiceDelivery
ServiceDelivery	DataReceivedAcknowledgement

NXTBUS supports the following **Publish/Subscribe** delivery protocols:

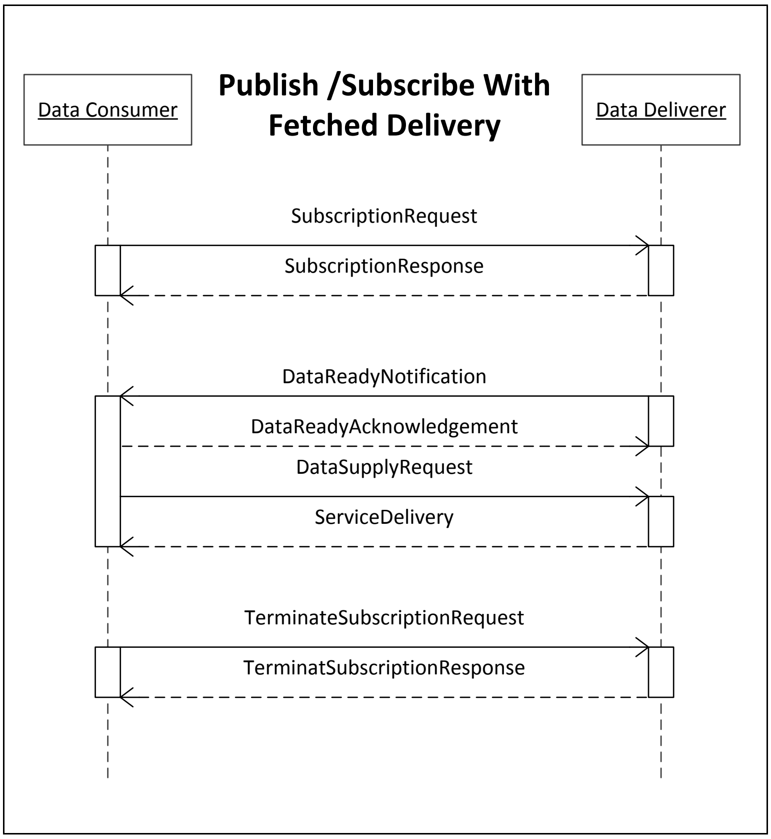
- Fetched Delivery, see 2.2.3.1
- Direct Delivery, see 2.2.3.2

NXTBUS requires the Subscriber and the Notification Consumer to be:

- Preconfigured: SubscriptionRequest.Address is not used.
- The Same: SubscriptionRequest.ConsumerAddress is not used.

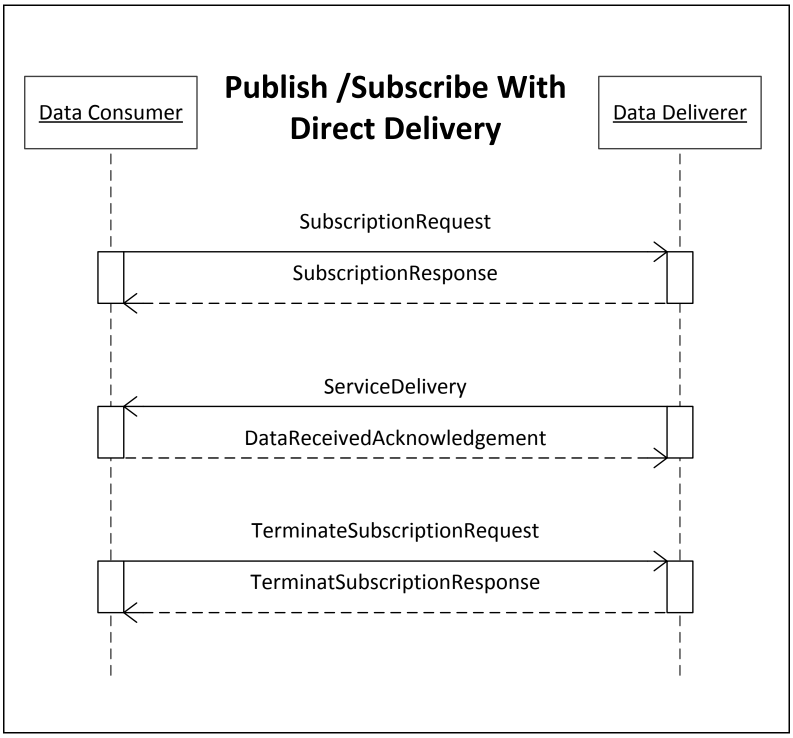
2.2.3.1 *Publish/Subscribe with fetched delivery*

Data is returned in two steps: first a DataReadyNotification message is sent from Producer to Consumer that data is available, and then a DataSupplyRequest message is sent from Consumer to Producer to get the data by a ServiceDelivery message.



2.2.3.2 *Publish/Subscribe with direct delivery*

Data is sent as the content of a single **ServiceDelivery** message to the DataConsumer (i.e. the notification and the delivery are the same message).


Publish /Subscribe With Direct Delivery
Data Consumer
Data Deliverer
SubscriptionRequest
SubscriptionResponse
ServiceDelivery
DataReceivedAcknowledgement
TerminateSubscriptionRequest
TerminatSubscriptionResponse

2.3 Accessing SIRI Data

2.3.1 Method of data exchange

NXTBUS exchanges messages via HTTP using the POST method, with an XML document containing the SIRI encoded message as a simple attachment.

All responses and subscriptions are sent to the address used by the requestor.

2.3.2 Formation of URLs

All requests must be directed at specific target URLs. The request URL depends on the service as well as the type of request. NXTBUS defines the following URL format:

HTTP_URL = http://{host}:{port}/{Requestor code}/{Service Type}/{Endpoint}

Parameter	NXTBUS Value
Host	nxtbus.act.gov.au
Port	11000
Requestor code	The API Key provided to the consumer. See 1.3.
Service Type	vm for Vehicle Monitoring sm for Stop Monitoring pt for Production Timetable et for Estimated Timetable
Endpoint	Service Endpoint, defined in 2.3.3 Endpoints

2.3.2.1 *Examples URLs*

Example of HTTP-requests to the stop monitoring service request/response service:

- http://siri.nxtbus.act.gov.au:11000/requestorcode/sm/service.xml

Examples of HTTP-requests to the stop monitoring publish/subscribe service from data consumer:

- CheckStatusRequest: http://siri.nxtbus.act.gov.au:11000/requestorcode/sm/status.xml
- SubscriptionRequest: http://siri.nxtbus.act.gov.au:11000/requestorcode/sm/subscription.xml
- DataSupplyRequest (Fetched Delivery): http://siri.nxtbus.act.gov.au:11000/requestorcode/sm/polldata.xml

Example of HTTP-requests to the stop monitoring publish/subscribe service from data deliverer:

- DataReadyNotification: http://siri.nxtbus.act.gov.au:11000/requestorcode/sm/dataready.xml

2.3.3 Endpoints

Name	Request code	Description
Poll status	status.xml	This request tests whether a service is responding.
Manage data subscription	subscription.xml	This request is used to retrieve online data from the desired control system or to delete existing subscriptions.
Report data ready	dataready.xml	This request reports on the readiness of data to a partner system. The partner system then initiates transmission of the data with a "transmit data" request.
Poll data	polldata.xml	This request is used to poll online data. As a reply, the

		ready data is transmitted or an error message.
Request for data	service.xml	ServiceRequest in Request/Response mode
Direct delivery	directdelivery.xml	ServiceDelivery in direct delivery mode

3 Request/Response

3.1 Production Timetable Request/Response

3.1.1 General

This service provides scheduled timetable information, usually the schedule of a whole operational day is exchanged in the early morning after operational day change.

The ProductionTimetableRequest should contain a ValidityPeriod with StartTime and EndTime, which defines the time range for which timetables have to be returned. If the Consumer requests for an EndTime later than the supported timetable-range (LIO-system: typically 6 o'clock at next morning), then the LIO-system sends:

- either a ProductionTimetableDelivery in Request/Response mode,
- or a SubscriptionResponse in Publish/Subscribe mode

with a ValidUntil = 6 o'clock at next morning.

The third party Consumer has to send a new ProductionTimetableRequest before the time in ValidUntil is reached.

3.1.2 Production Timetable Request

The following table identifies the elements which make up the ProductiontimetableRequest Message.

A ProductionTimetableRequest message is sent to the [Request for data] endpoint of a NXTBUS service.

Element	Type	Cardinality	Description
RequestTimestamp	xsd:dateTime	Required	Time stamp of the request.
ValidityPeriod		Optional	Two time points ('StartTime' and 'EndTime'), which establish the time window for the validity of the planning data. The times relate to the departure time at the start stop. As soon as the departure time from the start stop lies within the window (including the borders), the entire trip is transmitted in one piece.
ValidityPeriod.StartTime	xsd:datetime	Required	Start of the Time Window for the transmission of planning data.
ValidityPeriod.Endtime	xsd:datetime	Required	End of the Time Window for the transmission of planning data. If the end of

			a trip lies beyond the specified time window, information about the whole trip is transmitted anyway. Trapeze-specific implementation On the current day the subscriber is allowed to ask in maximum data until 05:00:00 of next day. If the partner subscribes for more data, he is informed by SubscriptionResponse.ResponseStatus.ValidUntil, that data are only available until 05:00:00 of next day.
Lines.LineDirection		Optional multiple	Filter, restricting the routes for which the planned schedules are to be requested.
Lines.LineDirection.LineRef	LineCode	Required	Filters results to the specified route.
Lines.LineDirection.DirectionRef	DirectionCode	Optional	Filters results to the specified direction.

Example of a **ProductionTimetableRequest** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb"
xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0"
xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <ServiceRequest>
    <RequestTimestamp>2013-12-29T12:58:29</RequestTimestamp>
    <RequestorRef>[Insert your API Key]</RequestorRef>
    <ProductionTimetableRequest version="2.0">
      <RequestTimestamp>2013-12-29T12:58:29</RequestTimestamp>
      <ValidityPeriod>
        <StartTime>2013-12-29T13:50:04+01:00</StartTime>
        <EndTime>2013-12-30T13:50:04+01:00</EndTime>
      </ValidityPeriod>
      <Lines>
        <LineDirection>
          <LineRef>ACT_200</LineRef>
          <DirectionRef>A</DirectionRef>
        </LineDirection>
      </Lines>
    </ProductionTimetableRequest>
  </ServiceRequest>

</Siri>
```

3.1.3 Production Timetable Response

A ProductionTimetableDelivery message is sent by the [Request for data] endpoint of a NXTBUS service in response to a ProductionTimetableRequest message.

Refer to chapter **5.2.3** and in particular **5.2.3.1** for a definition of the elements included in a ProductionTimetableDelivery message and an example of the message.

3.2 Stop Monitoring Request/Response

3.2.1 General

This service provides information about monitored visits to a given stop.

3.2.2 Stop Monitoring Request

The following table identifies the elements which make up the StopMonitoringRequest Message.

A StopMonitoringRequest message is sent to the [Request for data] endpoint of a NXTBUS service.

Element	Type	Cardinality	Description
RequestTimestamp	xsd:dateTime	Required	Time stamp of the request.
PreviewInterval	PositiveDurationType	Optional	Forward duration for which 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.
StartTime	xsd:dateTime	Optional	Start time for PreviewInterval. If absent, then current time is assumed
MonitoringRef	Monitoring-Code	Required	Reference to Monitoring Point(s) about which data is requested. Typically, this will be the code of the STOP POINT.
OperatorRef	OperatorCode	Optional	Filters results to the specified operator.
LineRef	LineCode	Optional	Filters results to the specified route.
DirectionRef	Direction-Code	Optional	Filters results to the specified direction.
DestinationRef	DestinationCode	Optional	Filters results to the specified destination.
StopVisitTypes	StopVisitTypeEnumeration	Optional	Filters results to the specified type of visit. Possible values are: • all • arrivals • departures
MaximumStopVisits	xsd:positive-Integer	Optional	• Maximum number of trips to return.

MaximumTextLenght	xsd:positive-Integer	Optional	Maximum length of text to return for text elements. Default is 30.

Example of a **StopMonitoringRequest** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="2.0"
xmlns="http://www.siri.org.uk/siri">
       <ServiceRequest>
              <RequestTimestamp>2014-02-24T09:58:42</RequestTimestamp>
              <RequestorRef>[Insert your API Key]</RequestorRef>
              <StopMonitoringRequest version="2.0">
                     <RequestTimestamp>2014-02-24T09:58:42</RequestTimestamp>
                     <PreviewInterval>PT30M</PreviewInterval>
                     <MonitoringRef>149000006105</MonitoringRef>
                     <MaximumStopVisits>4</MaximumStopVisits>
                     <MaximumTextLength>160</MaximumTextLength>
       </StopMonitoringRequest>
       </ServiceRequest>
</Siri>
```

3.2.3 Stop Monitoring Response

A StopMonitoringDelivery message is sent by the [Request for data] endpoint of a NXTBUS service in response to a StopMonitoringRequest message.

Refer to chapter **5.2.3.4** for a definition of the elements included in a StopMonitoringDelivery message.

Example of a **StopMonitoringDelivery** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.sir
i.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://ww
w.ifopt.org.uk/ifopt">
   <ServiceDelivery>
     <ResponseTimestamp>2013-12-30T17:44:03.000+01:00</ResponseTimestamp>
     <ProducerRef>VLOD</ProducerRef>
     <StopMonitoringDelivery version="2.0">
        <ResponseTimestamp>2013-12-30T17:44:03.000+01:00</ResponseTimestamp>
        <MonitoredStopVisit>
          <RecordedAtTime>2013-12-30T17:44:02.000+01:00</RecordedAtTime>
          <MonitoringRef>ANGB1B</MonitoringRef>
          <ClearDownRef>33</ClearDownRef>
          <MonitoredVehicleJourney>
             <LineRef>L1</LineRef>
             <DirectionRef>Ri B</DirectionRef>
             <FramedVehicleJourneyRef>
                <DataFrameRef>2013-12-30</DataFrameRef>
                <DatedVehicleJourneyRef>203821-00001-1</DatedVehicleJourneyRef>
             </FramedVehicleJourneyRef>
             <PublishedLineName>1</PublishedLineName>
             <DirectionName>Grubenstr.</DirectionName>
             <ExternalLineRef>1</ExternalLineRef>
             <OperatorRef>Erfurter Verkehrsbetriebe AG</OperatorRef>
             <ProductCategoryRef>Bus</ProductCategoryRef>
             <OriginName>Haupthof</OriginName>
             <Via>
                <PlaceName>1;Anger;2;Lutherkirche</PlaceName>
                <ViaPriority>1</ViaPriority>
             </Via>
             <DestinationName>Grubenstr.</DestinationName>
             <DestinationShortName>GRUS</DestinationShortName>
             <Monitored>true</Monitored>
             <InCongestion>false</InCongestion>
             <Occupancy>2</Occupancy>
             <BlockRef>100229</BlockRef>
             <CourseOfJourneyRef>59</CourseOfJourneyRef>
             <VehicleRef>469</VehicleRef>
             <MonitoredCall>
                <StopPointRef>ANGB_02</StopPointRef>
                <VisitNumber>9</VisitNumber>
                <VehicleAtStop>true</VehicleAtStop>
                <AimedArrivalTime>2013-12-30T17:38:00.000+01:00</AimedArrivalTime>
                <ExpectedArrivalTime>2013-12-
30T17:43:56.000+01:00</ExpectedArrivalTime>
                <AimedDepartureTime>2013-12-
30T17:38:00.000+01:00</AimedDepartureTime>
                <ExpectedDepartureTime>2013-12-
30T17:43:56.000+01:00</ExpectedDepartureTime>
                <DeparturePlatformName>Stg 2</DeparturePlatformName>
             </MonitoredCall>
          </MonitoredVehicleJourney>
        </MonitoredStopVisit>
     </StopMonitoringDelivery>
  </ServiceDelivery>
</Siri>
```

3.3 Vehicle Monitoring Request/Response

3.3.1 General

This service provides information about monitored vehicle activity.

3.3.2 Vehicle Monitoring Request

Element	Type	Cardinality	Description
RequestTimestamp	xsd:dateTime	Required	Time stamp of the request.
VehicleMonitoringRef	Monitoring-Code	Required	
LineRef	LineCode	Optional	Filters results to the specified route.

Element	Type	Cardinality	Description
DirectionRef	Direction-Code	Optional	Filters results to the specified direction.

Example of a **VehicleMonitoringRequest** message:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb"
xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0"
xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <ServiceRequest>
    <RequestTimestamp>2013-12-30T15:01:43.000+01:00</RequestTimestamp>
    <RequestorRef>[Insert your API Key]</RequestorRef>
    <VehicleMonitoringRequest version="2.0">
      <RequestTimestamp>2013-12-30T15:01:43.000+01:00</RequestTimestamp>
      <VehicleMonitoringRef>VM_ACT_0200</VehicleMonitoringRef>
    </VehicleMonitoringRequest>
  </ServiceRequest>

</Siri>
```

3.3.3 Vehicle Monitoring Response

A VehicleMonitoringDelivery message is sent by the [Request for data] endpoint of a NXTBUS service in response to a VehicleMonitoringRequest message.

Refer to chapter **5.2.3** and in particular **5.2.3.1** for a definition of the elements included in a VehicleMonitoringDelivery message and an example of the message.

4 Subscriptions

4.1 Setting up Subscriptions

4.1.1 General

A Subscription is created by sending a ***SubscriptionRequest*** to the Notification Producer of the desired SIRI Service type, as located by the [Manage data subscription] endpoint. The Notification Producer service responds with a ***SubscriptionResponse*** message that confirms the granting of the subscription, or provides an error condition that indicates why the subscription could not be created. The Notification Producer then creates the accepted subscriptions and starts to supply data.

4.1.2 Definition of SubscriptionRequest

The following table identifies the elements which make up the SubscriptionRequest Message.

A SubscriptionRequest message is sent to the [Manage data subscription] endpoint of a NXTBUS service.

Element	Type	Cardinality	Description
RequestTimestamp	xsd:dateTime	Required	Time stamp of the request.
RequestorRef	ParticipantRefStructure	Required	Requestor code of the requesting system.
ConsumerAddress		Optional	Address to which data is to be sent. Needed in Publish/Subscribe mode if the address cannot be determined by RequestorRef and preconfigured data.
SubscriptionContext. HeartbeatInterval	PositiveDurationType	Optional	The heartbeat interval may either be preconfigured for the whole system or specified as a parameter in the subscription request. If no HeartbeatInterval is received then a HeartbeatNotification is sent every 2 minutes.
ProductiontimetableSubscriptionRequest	ProductionTimetableSubscriptionRequest	Optional Multiple	This service provides information on the planned progress of vehicles operating a specific service, identified by the vehicle time of arrival and departure at specific stops on a planned route for a particular Operational Day.
EstimatedTimetableS	EstimatedTimetableS	Optional	This service provides information on the actual progress of vehicles

ubscriptionRequest	ubscriptionRequest	Multiple	operating a specific service detailing expected arrival and departure times at specific stops on a planned route. There will be recorded data for stops which have been passed, and predicted data for stops not yet passed. In addition the Estimated Timetable service allows Vehicle Journeys to be cancelled, added or changed.
VehicleMonitoring-SubscriptionRequest	VehicleMonitoring-SubscriptionRequest	Optional Multiple	This service provides information on the current location and status of identified vehicles.
StopMonitoring-SubscriptionRequest	StopMonitoring-SubscriptionRequest	Optional Multiple	This service provides details of arrivals and departures at a designated stop.

4.1.2.1 *Definition of EstimatedTimetableSubscriptionRequest*

Element	Type	Cardinality	Description
SubscriberRef	Participant-Code	Optional	Equal to RequestorRef
SubscriptionIdentifier	xsd:NMTOKEN	Required	A unique identifier specified by the displaying system.
InitialTerminationTime	xsd:dateTime	Required	Specifies the time until which the subscription is valid.
EstimatedTimeTableRequest.RequestTimestamp	xsd:datetime	Required	Timestamp of when the request was generated.
EstimatedTimetableRequest.PreviewInterval		Required	Maximum preview time in minutes requested by the schedule information system. To prevent the system from overload, the maximum PreviewInterval is set by Trapeze to 60 minutes, if a longer PreviewInterval is received.
Lines.LineDirection		Optional Multiple	Filter defining for which route real-time data is to be transmitted. No specification: Real-time data reported for all routes which are released for the subscribing partner.
ChangeBeforeUpdates		Required	Threshold value in seconds after which a deviation change is to be transmitted. To prevent the system from overload, the minimum hysteresis is set by Trapeze to 30s, if a smaller threshold is received.

Example of **EstimatedTimetableSubscriptionRequest** message:

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb"
xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0"
xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <SubscriptionRequest>
    <RequestTimestamp>2014-01-01T09:17:09</RequestTimestamp>
    <RequestorRef>[Insert your API Key]</RequestorRef>
    <EstimatedTimetableSubscriptionRequest>
      <SubscriberRef>[Insert your API Key]</SubscriberRef>
      <SubscriptionIdentifier>25</SubscriptionIdentifier>
      <InitialTerminationTime>2014-01-02T09:17:09.3755665Z</InitialTerminationTime>
      <EstimatedTimetableRequest version="2.0">
        <RequestTimestamp>2014-01-01T09:17:09</RequestTimestamp>
        <PreviewInterval>PT30M</PreviewInterval>
      </EstimatedTimetableRequest>
      <ChangeBeforeUpdates>PT30S</ChangeBeforeUpdates>
    </EstimatedTimetableSubscriptionRequest>
  </SubscriptionRequest>
</Siri>
```

4.1.2.2 *Definition of ProductionTimetableSubscriptionRequest*

Element	Type	Cardinality	Description
SubscriberRef	Participant-Code	Optional	Equal to RequestorRef
SubscriptionIdentifier	xsd:NMTOKEN	Required	A unique identifier specified by the displaying system.
InitialTerminationTime	xsd:dateTime	Required	Specifies the time until which the subscription is valid.
ProductionTimeTableRequest	ProductionTimetableRequest	Required	Production Timetable request definition

Example of **ProductionTimetableSubscriptionRequest** message:

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb"
xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0"
xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <SubscriptionRequest>
    <RequestTimestamp>2013-12-29T12:58:29</RequestTimestamp>
    <RequestorRef>[Insert your API Key]</RequestorRef>
    <ConsumerAddress>http://81.92.133.54:11002/zvv/et/directDelivery.xml
     </ConsumerAddress>
    <ProductionTimetableSubscriptionRequest>
      <SubscriberRef>[Insert your API Key]</SubscriberRef>
      <SubscriptionIdentifier>25</SubscriptionIdentifier>
      <InitialTerminationTime>2013-12-29T18:58:29.8689733Z</InitialTerminationTime>
      <ProductionTimetableRequest version="2.0">
        <RequestTimestamp>2013-12-29T12:58:29</RequestTimestamp>
        <ValidityPeriod>
          <StartTime>2013-12-29T13:50:04+01:00</StartTime>
          <EndTime>2013-12-30T13:50:04+01:00</EndTime>
        </ValidityPeriod>
      </ProductionTimetableRequest>
    </ProductionTimetableSubscriptionRequest>
  </SubscriptionRequest>
</Siri>
```

4.1.2.3 *Definition of VehicleMonitoringSubscriptionRequest*

Element	Type	Cardinality	Description
SubscriptionIdentifier	xsd:NMTOKEN	Required	A unique identifier specified by the displaying system.
InitialTerminationTime	xsd:dateTime	Required	Specifies the time until which the subscription is valid.
VehicleMonitoringRequest	VehicleMonitoringRequest	Required	Vehicle Monitoring Request data structure
UpdateInterval	PositiveDurationType	Required	Time interval in seconds in which new data is to be transmitted.

Example of a **VehicleMonitoringSubscriptionRequest** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <SubscriptionRequest>
    <RequestTimestamp>2013-12-30T15:01:43.000+01:00</RequestTimestamp>
    <RequestorRef>[Insert your API Key]</RequestorRef>
    <VehicleMonitoringSubscriptionRequest>
      <SubscriptionIdentifier>4</SubscriptionIdentifier>
      <InitialTerminationTime>2013-12-31T06:00:00.000+01:00</InitialTerminationTime>
      <VehicleMonitoringRequest version="2.0">
        <RequestTimestamp>2013-12-30T15:01:43.000+01:00</RequestTimestamp>
        <VehicleMonitoringRef>666</VehicleMonitoringRef>
      </VehicleMonitoringRequest>
      <UpdateInterval>P0Y0M0DT0H0M1.000S</UpdateInterval>
    </VehicleMonitoringSubscriptionRequest>
  </SubscriptionRequest>
</Siri>
```

4.1.2.4 *Definition of StopMonitoringSubscriptionRequest*

Element	Type	Cardinality	Description
SubscriberRef	Participant-Code	Optional	Equal to RequestorRef
SubscriptionIdentifier	*Subscription-Qualifier*	Required	A unique identifier specified by the displaying system.
InitialTerminationTime	xsd:dateTime	Required	Specifies the time until which the subscription is valid.

StopMonitoringRequest	StopMonitoringRequest	Required	Stop Monitoring Request data structure
ChangeBeforeUpdates	PositiveDurationType	Optional	Threshold value in seconds after which a deviation change is to be transmitted. [default=30min]

Example of a **StopMonitoringSubscriptionRequest** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.sir
i.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://ww
w.ifopt.org.uk/ifopt">
   <SubscriptionRequest>
      <RequestTimestamp>2013-12-30T15:01:43.000+01:00</RequestTimestamp>
      <RequestorRef>[Insert your API Key]</RequestorRef>
      <StopMonitoringSubscriptionRequest>
         <SubscriberRef>[Insert your API Key]</SubscriberRef>
         <SubscriptionIdentifier>4</SubscriptionIdentifier>
         <InitialTerminationTime>2013-12-
31T06:00:00.000+01:00</InitialTerminationTime>
         <StopMonitoringRequest version="2.0">
            <RequestTimestamp>2013-12-30T15:01:43.000+01:00</RequestTimestamp>
            <PreviewInterval>P0Y0M0DT1H0M0.000S</PreviewInterval>
            <MonitoringRef>GHP6B</MonitoringRef>
            <OperatorRef>1</OperatorRef>
            <LineRef>L6H</LineRef>
            <DirectionRef>L6RB</DirectionRef>
            <DestinationRef>L6RB</DestinationRef>
         </StopMonitoringRequest>
         <ChangeBeforeUpdates>P0Y0M0DT0H0M30.000S</ChangeBeforeUpdates>
      </StopMonitoringSubscriptionRequest>
   </SubscriptionRequest>
</Siri>
```

4.1.3 Definition of SubscriptionResponse

The following table identifies the elements which make up the SubscriptionResponse Message.

A SubscriptionResponse message is sent by the [Manage data subscription] endpoint of a NXTBUS service in response to a SubscriptionRequest message.

Element	Type	Cardinality	Description
ResponseTimestamp	xsd:dateTime	Required	Time stamp defining creation of the acknowledgement.
ResponderRef	ParticipantRefStructure	Optional	Participant reference that identifies responder.
ResponseStatus.ResponseTimestamp	xsd:dateTime	Required	Time stamp defining creation of the acknowledgement.

ResponseStatus.SubscriptionRef	xsd:NMTOKEN	Optional	Unique Identifier of Subscription.
ResponseStatus.Status	xsd:boolean	Optional	In good working case: ‘true’
(only sent in VehicleMonitoringDelivery): ResponseStatus.ShortestPossibleCycle	PositiveDurationType	Optional	Minimum interval at which updates can be sent.
(only sent in ProductionTimetable Delivery): ResponseStatus.ValidUntil		Optional	Expiry date for the data produced.

Example of a **SubscriptionResponse** message:

```
<?xml version=”1.0” encoding=”ISO-8859-1” standalone=”yes”?>
<Siri version=”2.0” xmlns:ns2=”http://www.ifopt.org.uk/acsb” xmlns=”http://www.sir
i.org.uk/siri” xmlns:ns4=”http://datex2.eu/schema/2_0RC1/2_0” xmlns:ns3=”http://ww
w.ifopt.org.uk/ifopt”>
  <SubscriptionResponse>
    <ResponseTimestamp>2014-01-01T10:17:09.000+01:00</ResponseTimestamp>
    <ResponderRef>VLOD</ResponderRef>
    <ResponseStatus>
      <ResponseTimestamp>2014-01-01T10:17:09.000+01:00</ResponseTimestamp>
      <SubscriptionRef>25</SubscriptionRef>
      <Status>true</Status>
    </ResponseStatus>
  </SubscriptionResponse>
</Siri>
```

4.2 Terminating Subscriptions

4.2.1 General

A Subscriber may terminate a subscription at any time by sending a TerminateSubscriptionRequest to the Subscription Manager.

4.2.2 Definition of TerminateSubscriptionRequest

The following table identifies the elements which make up the TerminateSubscriptionRequest Message.

A TerminateSubscriptionRequest message is sent to the [Manage data subscription] endpoint of a NXTBUS service.

Element	Type	Cardinality	Description
RequestTimestamp	xsd:dateTime	Required	Time stamp of generation of request
RequestorRef	ParticipantRefStructure	Required	Control centre code of the requesting system
All	EmptyType	Choice	All subscriptions have to be deleted.
SubscriptionRef		Choice	Set of subscriptions to be deleted.

Examples of a **TerminateSubscriptionRequest** message:

```
<?xml version=”1.0” encoding=”ISO-8859-1” standalone=”yes”?>
<Siri version=”2.0” xmlns:ns2=”http://www.ifopt.org.uk/acsb” xmlns=”http://www.sir
i.org.uk/siri” xmlns:ns4=”http://datex2.eu/schema/2_0RC1/2_0” xmlns:ns3=”http://ww
w.ifopt.org.uk/ifopt”>
  <TerminateSubscriptionRequest>
    <RequestTimestamp>2014-01-01T09:05:44</RequestTimestamp>
    <RequestorRef>[Insert your API Key]</RequestorRef>
    <All />
  </TerminateSubscriptionRequest>
</Siri>
```

```
<?xml version=”1.0” encoding=”ISO-8859-1” standalone=”yes”?>
<Siri version=”2.0” xmlns:ns2=”http://www.ifopt.org.uk/acsb” xmlns=”http://www.sir
i.org.uk/siri” xmlns:ns4=”http://datex2.eu/schema/2_0RC1/2_0” xmlns:ns3=”http://ww
w.ifopt.org.uk/ifopt”>
  <TerminateSubscriptionRequest>
    <RequestTimestamp>2014-01-01T09:15:43</RequestTimestamp>
    <RequestorRef>[Insert your API Key]</RequestorRef>
    <SubscriptionRef>32</SubscriptionRef>
  </TerminateSubscriptionRequest>
</Siri>
```

4.2.3 Definition of TerminateSubscriptionResponse

The following table identifies the elements which make up the TerminateSubscriptionResponse Message.

A TerminateSubscriptionResponse message is sent by the [Manage data subscription] endpoint of a NXTBUS service in response to a TerminateSubscriptionRequest message.

Element	Type	Cardinality	Description
ResponseTimestamp	xsd:dateTime	Required	Time stamp defining creation of the acknowledgement.
ResponderRef	ParticipantRefStructure	Optional	Participant reference that identifies responder.
TerminationResponseStatus.ResponseTime	xsd:dateTim	Required	Time stamp defining creation of the

stamp	e		acknowledgement.
TerminationResponseStatus.SubscriptionRef		Optional	Unique Identifier of Subscription.
TerminationResponseStatus.Status	xsd:boolean	Optional	In good working case: ‘true’

Example of a **TerminateSubscriptionResponse** message:

```
<?xml version=”1.0” encoding=”ISO-8859-1” standalone=”yes”?>
<Siri version=”2.0” xmlns:ns2=”http://www.ifopt.org.uk/acsb” xmlns=”http://www.sir
i.org.uk/siri” xmlns:ns4=”http://datex2.eu/schema/2_0RC1/2_0” xmlns:ns3=”http://ww
w.ifopt.org.uk/ifopt”>
  <TerminateSubscriptionResponse>
    <ResponseTimestamp>2014-01-01T10:05:44.000+01:00</ResponseTimestamp>
    <ResponderRef>VLOD</ResponderRef>
    <TerminationResponseStatus>
      <ResponseTimestamp>2014-01-01T10:05:44.000+01:00</ResponseTimestamp>
      <Status>true</Status>
    </TerminationResponseStatus>
  </TerminateSubscriptionResponse>
</Siri>
```

5 Delivering data

5.1 Fetched Data Delivery

5.1.1 General

(8.2.1. in ref. [2]) *Fetched Delivery* of subscriptions delivers the data in two steps; a ***DataReadyNotification*** from the Producer to the Consumer; and then a ***DataSupplyRequest*** from the Consumer to the Producer to fetch the data.

5.1.2 Definition of DataReadyNotification

The following table identifies the elements which make up the DataReadyNotification Message.

A DataReadyNotification message is sent by an endpoint of a NXTBUS service when there is data available for a subscription.

Element	Type	Cardinality	Description
RequestTimestamp	xsd:dateTime	Required	Time stamp of the notice of change message.
ProducerRef	ParticipantRefStructure	Optional	Control centre code of the data producing system.

Example of a **DataReadyNotification** message:

```
<?xml version=”1.0” encoding=”ISO-8859-1” standalone=”yes”?>
<Siri version=”2.0” xmlns:ns2=”http://www.ifopt.org.uk/acsb” xmlns=”http://www.sir
i.org.uk/siri” xmlns:ns4=”http://datex2.eu/schema/2_0RC1/2_0” xmlns:ns3=”http://ww
w.ifopt.org.uk/ifopt”>
   <DataReadyNotification>
      <RequestTimestamp>2014-01-01T12:10:36.000+01:00</RequestTimestamp>
      <ProducerRef>VLOD</ProducerRef>
   </DataReadyNotification>
</Siri>
```

5.1.3 Definition of DataReadyAcknowledgement

The following table identifies the elements which make up the DataReadyAcknowledgement message.

A DataReadyAcknowledgement message is sent to the endpoint of a NXTBUS service in response to a DataReadyNotification message.

. Element	Type	Cardinality	Description
ResponseTimestamp	xsd:dateTime	Required	Time stamp defining creation of the acknowledgement.
ConsumerRef	ParticipantRefStructure	Optional	Participant reference that identifies consumer.
Status	xsd:boolean	Optional	In good working case: ‘true’

Example of a **DataReadyAcknowledgement** message:

```
<?xml version=”1.0” encoding=”ISO-8859-1” standalone=”yes”?>
<Siri version=”2.0” xmlns:ns2=”http://www.ifopt.org.uk/acsb” xmlns=”http://www.sir
i.org.uk/siri” xmlns:ns4=”http://datex2.eu/schema/2_0RC1/2_0” xmlns:ns3=”http://ww
w.ifopt.org.uk/ifopt”>
   <DataReadyAcknowledgement>
      <ResponseTimestamp>2014-01-01T12:10:36+01:00</ResponseTimestamp>
      <ConsumerRef>HHA</ConsumerRef>
      <Status>true</Status>
   </DataReadyAcknowledgement>
</Siri>
```

5.2 Polling Data (DataSupplyRequest/ServiceDelivery)

5.2.1 Procedure

Data polling occurs at the request of the data Consumer. Only the current real-time information is transmitted. Historical data is not available.

Polling generally occurs after updated data is signaled by a DataReadyNotification, but can also occur at any time after setting up the subscription.

5.2.2 Definition of DataSupplyRequest

The following table identifies the elements which make up the DataSupplyRequest message.

A DataSupplyRequest message is sent to the [Poll Data] endpoint of a NXTBUS service.

If ***AllData*** is set to false, then the Producer will transmit only the data that has been updated since the last request. The read is destructive; that is, it cannot be repeated because once the data has been fetched the last update flag has been reset.

If ***AllData*** is set to true, then the Producer will transmit not only the data that has been updated since the last request, but all data for all active subscriptions held by the Consumer. The read is non-destructive; that is, it can be repeated until the data is stale.

Element	Type	Cardinality	Description
RequestTimestamp	xsd:dateTime	Required	Time stamp of generation of request
ConsumerRef	ParticipantRefStructure	Optional	Control centre code of the consumer requesting data.
AllData	xsd:boolean	Optional	Whether to return all data, or just new updates since the last delivery. Default value: false

Example of a **DataSupplyRequest** message:

```
<?xml version=”1.0” encoding=”ISO-8859-1” standalone=”yes”?>
<Siri version=”2.0” xmlns:ns2=”http://www.ifopt.org.uk/acsb” xmlns=”http://www.sir
i.org.uk/siri” xmlns:ns4=”http://datex2.eu/schema/2_0RC1/2_0” xmlns:ns3=”http://ww
w.ifopt.org.uk/ifopt”>
   <DataSupplyRequest>
      <RequestTimestamp>2013-12-29T13:58:30+01:00</RequestTimestamp>
      <ConsumerRef>HAFAS</ConsumerRef>
      <AllData>false</AllData>
   </DataSupplyRequest>
</Siri>
```

5.2.3 Definition of ServiceDelivery

The following table identifies the elements which make up the ServiceDelivery message.

A ServiceDelivery message is sent by the [Poll Data] endpoint of a NXTBUS service in response to a DataSupplyRequest message.

Element	Type	Cardinality	Description
ResponseTimestamp	xsd:dateTime	Required	Time stamp defining creation of the acknowledgement.
ProducerRef	ParticipantRef Structure	Optional	Participant reference that identifies producer of data. May be available from context.
Status	xsd:boolean	Optional	‘true’ in good working case
MoreData	xsd:boolean	Optional	‘true’ if other data has to be polled, otherwise “false” (default).
ProductionTimetable Delivery	ProductionTimetableDelivery	Optional Multiple	Contains published timetable information.
EstimatedTimetableDelivery	EstimatedTime tableDelivery	Optional	Contains actual schedule information for a vehicle on a planned route.

		Multiple	
StopMonitoringDelivery	StopMonitoringDelivery	Optional Multiple Choice	Contains predicted arrivals/departures of third party vehicles approaching a display area, see service-specific document for more details.
VehicleMonitoringDelivery	VehicleMonitoringDelivery	Optional Multiple Choice	Contains information about trips that are to be visualised in a third party control centre, see service-specific document for more details.

5.2.3.1 *Definition of ProductionTimetableDelivery*

Element	Type	Cardinality	Description
version	VersionString	Required	Version identifier of Vehicle Monitoring Service. Fix set to ‘2.0’.
ResponseTimestamp	xsd:dateTime	Required	Time individual response element was created.
SubscriptionRef	ParticipantRefStructure	Required	The SubscriptionRef references the subscription of visualisation data created by the request. The SubscriptionRef is specified by the displaying system.
ValidUntil	xsd:datetime	Optional	End of data horizon of the data producer. Has to be sent when not the whole subscribed ValidityPeriod is available.
DatedTimetableVersionFrame	DatedTimetableVersionFrame	Optional Multiple	Contains the planned schedule of a route in a specific direction.

5.2.3.1.1 Definition of DatedTimetableVersionFrame

Element	Type	Cardinality	Description
RecordedAtTime	xsd:datetime	Required	Time stamp of generation of DatedTimetableVersionFrame.

VersionRef		Optional	Timetable Version. Active version of data supply.
LineRef		Required	
DirectionRef		Required	
PublishedLineName	NaturalLanguageStringStructure	Optional	Name or Number by which the route is known to the public. Can be overwritten by the PublishedLineName in the DatedVehicleJourney structure.
ProductCategoryRef		Optional	
DestinationDisplay	NaturalLanguageStringStructure	Optional	Name of the destination the journey is going to, usually the same text as displayed on the signs or on the front of the vehicle. The mostly used DirectionName in the journeys of every route-direction-combination is filled. Can be overwritten by the DirectionName in the DatedVehicleJourney structure.
DatedVehicleJourney	DatedVehicleJourney	Optional Multiple	Provides schedule information about the vehicle journey along which a vehicle is running.

5.2.3.1.2 Definition of DatedVehicleJourney

Element	Type	Cardinality	Description
DatedJourneyVehicleCode		Required	Unique reference to the trip depending on the operational day.
PublishedLineName		Optional	Name or Number by which the route is known to the public. Only sent if different from the one that is sent in DatedTimetableVersionFrame.
VehicleFeatureRef		Optional Multiple	Feature(s) of the vehicle providing the journey.
DestinationDisplay	NaturalLanguageStringStructure	Optional	Name of the destination the journey is going to (DpiDestinationText of end stop or name of last productive stop), usually the same text as displayed on the signs or on the front of the vehicle. Only sent if different to the DirectionName sent in DatedTimetableVersionFrame. Can be overwritten section by section by the DestinationDisplay in the

			DatedCall structure.
DatedCalls.DatedCall	DatedCall	Optional Multiple	Complete sequence of stops along the pattern, in calling order, with details of every stop.

5.2.3.1.3 Definition of DatedCall

Element	Type	Cardinality	Description
StopPointRef	xsd:NMTOKEN	Required	Reference to a SCHEDULED STOP POINT. Defaults to that of context i.e. that specified on ***MonitoredStopVisit***.
DestinationDisplay	NaturalLanguageStringStructure	Optional	DpiDestinationText or StopName of the next DPI relevant intermediate stop the journey is going to, usually the same text as displayed on the signs or on the front of the vehicle. Filled because it's different to the DirectionName sent in DatedVehicleJourney structure.
AimedArrivalTime	xsd:datetime	Optional	Planned arrival time at the stop
ArrivalPlatformName	NaturalLanguageStringStructure	Optional	Name of bay or platform at which the vehicle will arrive
AimedDepartureTime	xsd:datetime	Optional	Planned departure time from the stop
DeparturePlatformName	NaturalLanguageStringStructure	Optional	Name of bay or platform from which vehicle will depart.
DepartureBoardingActivity		Optional	Type of boarding allowed at stop. Possible values: • 'boarding' • 'noBoarding' • 'passThru' • not sent means the default 'boarding

Example of **ProductionTimetableDelivery** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb"
xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0"
xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <ServiceDelivery>
    <ResponseTimestamp>2013-12-29T13:58:30.000+01:00</ResponseTimestamp>
    <ProducerRef>ACT</ProducerRef>
    <Status>true</Status>
    <ProductionTimetableDelivery version="2.0">
      <ResponseTimestamp>2013-12-29T13:58:30.000+01:00</ResponseTimestamp>
      <SubscriptionRef>25</SubscriptionRef>
      <ValidUntil>2013-12-30T06:00:00.000+01:00</ValidUntil>
      <DatedTimetableVersionFrame>
        <RecordedAtTime>2013-12-29T13:58:30.000+01:00</RecordedAtTime>
        <VersionRef>71091005</VersionRef>
        <LineRef>L1</LineRef>
        <DirectionRef>Ri A</DirectionRef>
        <PublishedLineName>1</PublishedLineName>
        <ProductCategoryRef>Bus</ProductCategoryRef>
        <VehicleFeatureRef>RENAULT PR100-3</VehicleFeatureRef>
        <VehicleFeatureRef>Wheelchair Accessible Bus</VehicleFeatureRef>
        <VehicleFeatureRef>Single Deck Bus</VehicleFeatureRef>
        <DestinationDisplay>Haupthof</DestinationDisplay>
        <DatedVehicleJourney>
          <DatedVehicleJourneyCode>2476-00004-1-20131229</DatedVehicleJourneyCode>
          <DatedCalls>
            <DatedCall>
              <StopPointRef>GRUS_04</StopPointRef>
              <AimedDepartureTime>2013-12-29T14:03:30.000+01:00</AimedDepartureTime>
            </DatedCall>
            <DatedCall>
              <StopPointRef>GRUS_06</StopPointRef>
              <AimedArrivalTime>2013-12-29T14:04:00.000+01:00</AimedArrivalTime>
              <AimedDepartureTime>2013-12-29T14:04:00.000+01:00</AimedDepartureTime>
            </DatedCall>
              :
              :
            <DatedCall>
              <StopPointRef>5801</StopPointRef>
              <AimedArrivalTime>2013-12-29T14:28:00.000+01:00</AimedArrivalTime>
              <AimedDepartureTime>2013-12-29T14:28:00.000+01:00</AimedDepartureTime>
            </DatedCall>
            <DatedCall>
              <StopPointRef>5802</StopPointRef>
              <AimedArrivalTime>2013-12-29T14:29:00.000+01:00</AimedArrivalTime>
            </DatedCall>
          </DatedCalls>
        </DatedVehicleJourney>
      </DatedTimetableVersionFrame>
    </ProductionTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

5.2.3.2 *Definition of EstimatedTimetableDelivery*

Element	Type	Cardinality	Description
version	VersionString	Required	Version identifier of Vehicle Monitoring Service. Fix set to ‘2.0’.
ResponseTimestamp	xsd:dateTime	Required	Time individual response element was created.
SubscriptionRef	ParticipantRefStruct	Required	The SubscriptionRef references

	ure		the subscription of visualisation data created by the request. The SubscriptionRef is specified by the displaying system.
EstimatedJourneyVersionFrame.RecordedAtTime	xsd:datetime	Required	Time that EstimatedJourneyVersionFrame was generated.
EstimatedJourneyVersionFrame.EstimatedVehicleJourney	EstimatedVehicleJourney	Required Multiple	Actual information about the trip

5.2.3.2.1 Definition of EstimatedVehicleJourney

Element	Type	Cardinality	Description
LineRef		Required	The LineRef is used to match the AVLC route with the route of the long-term schedule known within the schedule information service.
DirectionRef		Required	Reference to the relative direction the vehicle is running along the pattern, for example, ‘in’ or ‘out.
DatedVehicleJourneyRef		Choice	
EstimatedVehicleJourneyCode		Choice	
ExtraJourney	xsd:boolean	Optional	‘true’ indicates that this trip is transmitted as an addition to the planning data. If this element is not sent, then the default ‘false’ is taken.
Cancellation	xsd:Boolean	Optional	‘true’ indicates that this trip is cancelled. If this element is not sent, then the default ‘false’ is taken.
PublishedLineName		Optional	Name or Number by which the route is known to the public.
SIRI v1.3: DirectionName **SIRI v2.0: DestinationDisplayAtOrigin (RichtungsText)**		Optional	Name of the destination the vehicle is running to (DpiDestinationText of end stop or name of last productive stop), usually the same text as displayed on the signs or on the front of the vehicle. Can be overwritten section by section by the DestinationDisplay in the EstimatedCall structure.
ProductCategoryRef		Optional	Product category of journey, e.g. ‘Bus’, ‘Rail’, ‘Tram’.
VehicleFeatureRef		Optional Multiple	Features of the vehicle providing the journey.

JourneyNote	NaturalLanguageStringStructure	Optional	Information on this trip. Used to transmit: • either a TripSpecialText set up by the dispatcher for the whole trip (TripSpecialText > TripSpecialText, not TripSpecialText > DestinationText) • or a RouteSpecialText.
Monitored	xsd:boolean	Optional	'true' indicates that a vehicle is logged on, 'false' indicates that no vehicle is logged on (anymore).
PredictionInaccurate	xsd:boolean	Optional	'true' indicates that the prediction is inaccurate. If this element is not sent, then the default 'false' is taken.
Occupancy		Optional	
EstimatedCalls.EstimatedCall	EstimatedCall	Optional Multiple	Depending on IsCompleteStopSequence: • either complete sequence • or list of stops along the pattern where the deviation changes more than the subscribed hysteresis, in calling order.
IsCompleteStopSequence	xsd:boolean	Optional	'true' indicates that all stop passages for the trip are sent in the message. All stops are sent in many different cases, but general rule is that if the trip structure has changed then it is the trigger for sending all stop passages again, because partner agency must be always aware about the current structure of the trip. 'false' indicates that the 'Supplementary Rule' is used.

5.2.3.2.2 Definition of EstimatedCall

Element	Type	Cardinality	Description
StopPointRef		Optional	
ExtraCall	xsd:Boolean	Optional	'true' indicates that this is an additional stop passage for this journey. If this element is not sent, then the default 'false' is taken.
DestinationDisplay	NaturalLanguageStri	Optional	DpiDestinationText or StopName of

	ngStructure		the next DPI relevant intermediate stop the vehicle is running to, usually the same text as displayed on the signs or on the front of the vehicle. Filled because it's different to the DirectionName in EstimatedVehicleJourney structure.
CallNote	NaturalLanguageStringStructure	Optional	Trip Special Text that only applies at this stop.
AimedArrivalTime	xsd:datetime	Optional	Planned arrival time provided by schedule service.
ExpectedArrivalTime	xsd:datetime	Optional	Predicted arrival time provided by schedule service. When not set, it means no predicted arrival time known for this stop.
ArrivalPlatformName		Optional	Bay or name of arrival platform.
AimedDepartureTime	xsd:datetime	Optional	Planned departure time provided by schedule service.
ExpectedDepartureTime	xsd:datetime	Optional	Predicted departure time provided by schedule service. When not set, it means no predicted departure time known for this stop.
DeparturePlatformName		Optional	Bay or name of departure platform.
DepartureBoardingActivity	xsd:boolean	Optional	'true' indicates that passengers cannot embark at this stop passage. If this element is not sent, then the default 'false' is taken.

Example of **EstimatedTimetableDelivery** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb"
xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0"
xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <ServiceDelivery>
    <ResponseTimestamp>2013-12-31T18:24:18.000+01:00</ResponseTimestamp>
    <ProducerRef>ACT</ProducerRef>
    <Status>true</Status>
    <EstimatedTimetableDelivery version="2.0">
      <ResponseTimestamp>2013-12-31T18:24:16.000+01:00</ResponseTimestamp>
      <SubscriptionRef>25</SubscriptionRef>
      <EstimatedJourneyVersionFrame>
        <RecordedAtTime>2013-12-31T18:24:18.000+01:00</RecordedAtTime>
        <EstimatedVehicleJourney>
          <LineRef>L61</LineRef>
          <DirectionRef>Ri B</DirectionRef>

          <DatedVehicleJourneyRef>163921-00001-1-20131231</DatedVehicleJourneyRef> or
          <EstimatedVehicleJourneyRef>163921-00001-2-20131231
</EstimatedVehicleJourneyRef>

          <ExtraJourney>true</ExtraJourney>
          <Cancellation>false</Cancellation>
          <PublishedLineName>D61</PublishedLineName>
          <DestinationDisplayAtOrigin>Grubenstr.</DestinationDisplayAtOrigin>
          <ProductCategoryRef>Bus</ProductCategoryRef>
          <VehicleFeatureRef>lowFloor</VehicleFeatureRef>
          <VehicleFeatureRef>Wheelchair Accessible Bus</VehicleFeatureRef>
          <VehicleFeatureRef>Single Deck Bus</VehicleFeatureRef>
          <JourneyNote>Journey is shortened due to the accident at the University
Bridge</JourneyNote>
          <Monitored>true</Monitored>
          <PredictionInaccurate>false</PredictionInaccurate>
          <Occupancy>2</Occupancy>
          <EstimatedCalls>
            <EstimatedCall>
              <StopPointRef>5802</StopPointRef>
              <ExtraCall>false</ExtraCall>
              <CallNote>Possible delays due to the accident at the University
Bridge</CallNote>
              <AimedDepartureTime>2013-12-31T18:54:00.000+01:00</AimedDepartureTime>
              <ExpectedDepartureTime>2013-12-
31T18:54:00.000+01:00</ExpectedDepartureTime>
            </EstimatedCall>
            <EstimatedCall>
              <StopPointRef>5302</StopPointRef>
              <ExtraCall>false</ExtraCall>
              <CallNote>Possible delays due to the accident at the University
Bridge</CallNote>
              <AimedArrivalTime>2013-12-31T18:55:00.000+01:00</AimedArrivalTime>
              <ExpectedArrivalTime>2013-12-31T18:55:00.000+01:00</ExpectedArrivalTime>
              <ArrivalPlatformName>A</ArrivalPlatformName>
              <AimedDepartureTime>2013-12-31T18:55:00.000+01:00</AimedDepartureTime>
              <ExpectedDepartureTime>2013-12-
31T18:55:00.000+01:00</ExpectedDepartureTime>
              <DeparturePlatformName>A</DeparturePlatformName>
            </EstimatedCall>
            :
            <EstimatedCall>
              <StopPointRef>1703</StopPointRef>
              <ExtraCall>false</ExtraCall>
              <CallNote>Possible delays due to the accident at the University
Bridge</CallNote>
              <AimedArrivalTime>2013-12-31T19:17:00.000+01:00</AimedArrivalTime>
              <ExpectedArrivalTime>2013-12-31T19:17:00.000+01:00</ExpectedArrivalTime>
              <AimedDepartureTime>2013-12-31T19:17:00.000+01:00</AimedDepartureTime>
              <ExpectedDepartureTime>2013-12-
31T19:17:00.000+01:00</ExpectedDepartureTime>
            </EstimatedCall>
            <EstimatedCall>
              <StopPointRef>GRUS_05</StopPointRef>
              <ExtraCall>false</ExtraCall>
              <CallNote>Possible delays due to the accident at the University
Bridge</CallNote>
              <AimedArrivalTime>2013-12-31T19:18:00.000+01:00</AimedArrivalTime>
              <ExpectedArrivalTime>2013-12-31T19:18:00.000+01:00</ExpectedArrivalTime>
              <DepartureBoardingActivity>true</DepartureBoardingActivity>
            </EstimatedCall>
          </EstimatedCalls>
          <IsCompleteStopSequence>true</IsCompleteStopSequence>
        </EstimatedVehicleJourney>
      </EstimatedJourneyVersionFrame>
    </EstimatedTimetableDelivery>
```

```
  </ServiceDelivery>
</Siri>
```

5.2.3.3 *Definition of VehicleMonitoringDelivery*

Element	Type	Cardinality	Description
version	VersionString	Required	Version identifier of Vehicle Monitoring Service. Fix set to '2.0'.
ResponseTimestamp	xsd:dateTime	Required	Time individual response element was created.
SubscriptionRef	ParticipantRefStructure	Required	The SubscriptionRef references the subscription of visualisation data created by the request. The SubscriptionRef is specified by the displaying system.
VehicleActivity	VehicleActivity	Optional Multiple	Describes the progress of a VEHICLE along its route.
VehicleActivityCancellation	VehicleActivityCancellation	Optional Multiple	Reference to a previously communicated VehicleActivity which has to be deleted again.

5.2.3.3.1 Definition of VehicleActivity

Element	Type	Cardinality	Description
RecordedAtTime	*xsd:dateTime*	Required	Time at which VEHICLE data was recorded.
ValidUntilTime	xsd:dateTime	Required	Time until which data is valid.
VehicleMonitoringRef	*VehicleMonitoring-Identifier*	Required	Reference to a Vehicle Monitoring scope.
ProgressBetweenStops.LinkDistance	xsd:decimal	Optional	The total distance in metres between the previous stop and the next stop.
MonitoredVehicleJourney.LineRef	LineCode	Required	Reference to a LINE.
MonitoredVehicleJourney.DirectionRef	DirectionCode	Required	Reference to a DIRECTION the VEHICLE is running along the LINE, for example, 'in' or 'out', 'clockwise'. Distinct from a DESTINATION.
MonitoredVehicleJourney.	xsd:NMTOKE	Required	Unique identifier of data frame

FramedVehicleJourneyRef .DataFrameRef	N		within participant service. Used to ensure that the ***DatedVehicleJourneyRef*** is unique with the data range of the producer. Often the ***OperationalDayType*** is used for this purpose.
MonitoredVehicleJourney. FramedVehicleJourneyRef .DatedVehicleJourneyRef	xsd:NMTOKEN	Required	A reference to the DATED VEHICLE JOURNEY that the VEHICLE is making.
MonitoredVehicleJourney. JourneyPatternRef	xsd:NMTOKEN	Optional	Reference to the pattern along which the vehicle is driving.
MonitoredVehicleJourney. PublishedLineName	NaturalLanguageStringStructure	Optional	Name or Number by which the LINE is known to the public.
MonitoredVehicleJourney. DirectionName	NaturalLanguageStringStructure	Optional	Name of the relative direction the VEHICLE is running along the LINE, for example, "inbound" or "outbound”.
MonitoredVehicleJourney. ExternalLineRef	xsd:NMTOKEN	Optional	Alternative identifier of LINE that an external system may associate with journey.
MonitoredVehicleJourney. OperatorRef	xsd:NMTOKEN	Optional	OPERATOR of journey.
MonitoredVehicleJourney. ProductCategoryRef	ProductCategoryRefStructure	Optional	PRODUCT CATEGORY of journey – classifies, for example; express, local.
MonitoredVehicleJourney. OriginName	NaturalLanguagePlaceNameStructure	Optional	The name of the origin of the journey; used to help identify the VEHICLE to the public.
MonitoredVehicleJourney. OriginShortName	NaturalLanguagePlaceNameStructure	Optional	The short name of the origin of the journey; used to help identify the VEHICLE to the public.
MonitoredVehicleJourney. DestinationName	NaturalLanguagePlaceNameStructure	Optional	The name of the destination of the journey; used to help identify the VEHICLE to the public.
MonitoredVehicleJourney. DestinationShortName	NaturalLanguagePlaceNameStructure	Optional	The name of the destination of the journey; used to help identify the VEHICLE to the public.
MonitoredVehicleJourney. OriginAimedDepartureTime	xsd:dateTime	Optional	Timetabled departure time of VEHICLE from Origin.
MonitoredVehicleJourney. DestinationAimedArrivalTime	xsd:dateTime	Optional	Timetabled arrival time of VEHICLE at Destination.

MonitoredVehicleJourney.Monitored	xsd:boolean	Optional	Whether there is real-time information available for journey, if not present, not known.
MonitoredVehicleJourney.InCongestion	xsd:boolean	Optional	Whether the vehicle is in congestion. If not, present, not known.
MonitoredVehicleJourney.VehicleLocation.Longitude	xs:float	Optional	Longitude from Greenwich Meridian. 180° (East) to +180° (West). Decimal degrees. e.g. 2.356.
MonitoredVehicleJourney.VehicleLocation.Latitude	xs:float	Optional	Latitude from equator. -90° (South) to +90° (North). Decimal degrees. e.g. 56.356.
MonitoredVehicleJourney.Occupancy	Occupancy level code	Optional	Indicates how full the vehicle is. May be one of the values '1' / '2' / '3' / '4' (passenger load level determined by LIO-System, '4' means overloaded).
MonitoredVehicleJourney.Delay	DurationType	Optional	Delay to a precision in seconds. Early times are shown as negative values.
MonitoredVehicleJourney.BlockRef	xsd:NMTOKEN	Optional	BLOCK that VEHICLE is running.
MonitoredVehicleJourney.CourseOfJourneyRef	xsd:NMTOKEN	Optional	Run that VEHICLE is running.
MonitoredVehicleJourney.VehicleRef	xsd:NMTOKEN	Optional	A reference to the specific VEHICLE making a journey.
MonitoredVehicleJourney.MonitoredCall.StopPointRef	xsd:NMTOKEN	Optional	Reference to a SCHEDULED STOP POINT. Defaults to that of context i.e. that specified on ***MonitoredStopVisit***.
MonitoredVehicleJourney.MonitoredCall.VisitNumber	xsd:positiveInteger	Optional	For JOURNEY PATTERNs that involve repeated visits by a VEHICLE to the same stop, the ***VisitNumber*** is used to distinguish each separate visit.
MonitoredVehicleJourney.MonitoredCall.VehicleAtStop	xsd:boolean	Optional	Whether VEHICLE is at stop at the current time. If absent, unknown.
MonitoredVehicleJourney.OnwardCalls.OnwardCall.StopPointRef	xsd:NMTOKEN	Optional	SIRI: Reference to next SCHEDULED STOP POINT. Defaults to that of context i.e. that specified on ***MonitoredStopVisit***.

Example of a **VehicleMonitoringDelivery** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
```

```
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://www.ifopt.org.uk/ifopt">
   <ServiceDelivery>
      <ResponseTimestamp>2013-12-29T17:57:31.000+01:00</ResponseTimestamp>
      <ProducerRef>VLOD</ProducerRef>
      <Status>true</Status>
      <VehicleMonitoringDelivery version="2.0">
         <ResponseTimestamp>2013-12-29T17:57:31.000+01:00</ResponseTimestamp>
         <SubscriptionRef>26</SubscriptionRef>
         <ValidUntil>2013-12-29T18:07:23.000+01:00</ValidUntil>
         <VehicleActivity>
            <RecordedAtTime>2013-12-29T17:57:22.000+01:00</RecordedAtTime>
            <ValidUntilTime>2013-12-29T18:07:22.000+01:00</ValidUntilTime>
            <VehicleMonitoringRef>111</VehicleMonitoringRef>
            <ProgressBetweenStops>
               <Percentage>0</Percentage>
            </ProgressBetweenStops>
            <MonitoredVehicleJourney>
               <LineRef>L61</LineRef>
               <DirectionRef>Ri B</DirectionRef>
               <FramedVehicleJourneyRef>
                  <DataFrameRef>2013-12-29</DataFrameRef>
                  <DatedVehicleJourneyRef>243091-00004-1</DatedVehicleJourneyRef>
               </FramedVehicleJourneyRef>
               <JourneyPatternRef>9002</JourneyPatternRef>
               <PublishedLineName>D61</PublishedLineName>
               <DirectionName>Grubenstr.</DirectionName>
               <ExternalLineRef>61</ExternalLineRef>
               <OperatorRef>Erfurter Verkehrsbetriebe AG</OperatorRef>
               <ProductCategoryRef>Bus</ProductCategoryRef>
               <Monitored>true</Monitored>
               <InCongestion>true</InCongestion>
               <VehicleLocation>
                  <Longitude>180.0</Longitude>
                  <Latitude>90.0</Latitude>
               </VehicleLocation>
               <Occupancy>2</Occupancy>
               <Delay>-P0Y0M0DT0H2M50.000S</Delay>
               <BlockRef>6102</BlockRef>
               <CourseOfJourneyRef>2</CourseOfJourneyRef>
               <VehicleRef>435</VehicleRef>
               <MonitoredCall>
                  <StopPointRef>GRUS_04</StopPointRef>
                  <VehicleAtStop>true</VehicleAtStop>
               </MonitoredCall>
               <OnwardCalls>
                  <OnwardCall/>
               </OnwardCalls>
            </MonitoredVehicleJourney>
         </VehicleActivity>
      </VehicleMonitoringDelivery>
   </ServiceDelivery>
</Siri>
```

5.2.3.3.2 Definition of VehicleActivityCancellation

Element	Type	Cardinality	Description
RecordedAtTime	*xsd:dateTime*	Required	Time at which VEHICLE data was recorded.
VehicleMonitoringRef	*VehicleMonitoring-Code*	Required	Reference to a Vehicle Monitoring scope.
VehicleJourneyRef.DataFrameRef		Required	Unique identifier of data frame within participant service. Used to ensure that the ***DatedVehicleJourneyRef*** is unique with the data range of the producer. Often the ***OperationalDayType*** is used for this purpose.
VehicleJourneyRef.DatedVehicleJourneyRef		Required	A reference to the DATED VEHICLE JOURNEY that the VEHICLE is making.
LineRef	*LineCode*	Required	Reference to a LINE.
DirectionRef	*DirectionCode*	Required	Reference to a DIRECTION the VEHICLE is running along the LINE, for example, ‘in’ or ‘out’, ‘clockwise’. Distinct from a DESTINATION.
PublishedLineName		Optional	Name or Number by which the LINE is known to the public.
DirectionName		Optional	Name of the relative direction the VEHICLE is running along the LINE, for example, ‘inbound’ or ‘outbound’.
ExternalLineRef		Optional	Alternative identifier of LINE that an external system may associate with journey.
Reason	*NLString*	Optional	Reason for the cancellation of the VehicleActivity.

Example of a **VehicleActivityCancellation** message:

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <ServiceDelivery>
    <ResponseTimestamp>2013-12-29T19:53:30.000+01:00</ResponseTimestamp>
    <ProducerRef>VLOD</ProducerRef>
    <Status>true</Status>
    <VehicleMonitoringDelivery version="2.0">
```

```
         <ResponseTimestamp>2013-12-29T19:53:30.000+01:00</ResponseTimestamp>
         <SubscriptionRef>32</SubscriptionRef>
         <ValidUntil>2013-12-29T20:03:02.000+01:00</ValidUntil>
         <VehicleActivityCancellation>
            <RecordedAtTime>2013-12-29T19:53:02.000+01:00</RecordedAtTime>
            <VehicleMonitoringRef>VVLOD006</VehicleMonitoringRef>
            <VehicleJourneyRef>
               <DataFrameRef>2013-12-29</DataFrameRef>
               <DatedVehicleJourneyRef>5067-00004-1</DatedVehicleJourneyRef>
            </VehicleJourneyRef>
            <LineRef>L6</LineRef>
            <DirectionRef>Ri B</DirectionRef>
            <PublishedLineName>6</PublishedLineName>
            <DirectionName>Windischh.</DirectionName>
            <ExternalLineRef>6</ExternalLineRef>
            <Reason>trip finished</Reason>
         </VehicleActivityCancellation>
      </VehicleMonitoringDelivery>
   </ServiceDelivery>
</Siri>
```

5.2.3.4 *Definition of StopMonitoringDelivery*

Element	Type	Cardinality	Description
version	*VersionString*	Required	Version identifier of Stop Monitoring Service. Fixed set to ‘2.0’.
ResponseTimestamp		Required	Time individual response element was created.
SubscriptionRef		Required	The SubscriptionRef references the subscription of visualisation data created by the request. The SubscriptionRef is specified by the displaying system.
MonitoredStopVisit	MonitoredStopVisit	Optional Multiple	A visit to a SCHEDULED STOP POINT by a VEHICLE as an arrival and /or departure.
MonitoredStopVisitCancellation	MonitoredStopVisitCancellation	Optional Multiple	Reference to an previously communicated which should now be removed from the arrival/departure board for the stop.
StopLineNotice	StopLineNotice	Optional Multiple	A notice concerning a LINE.
StopLineNotice Cancellation	StopLineNoticeCancellation	Optional Multiple	Reference to a previously communicated StopLineNotice which should now be removed from the arrival/departure board for the stop.

5.2.3.4.1 Definition of MonitoredStopVisit

Element	Type	Cardinality	Description
RecordedAtTime	*xsd:dateTime*	Required	Time at which data was recorded.
ValidUntilTime		Required	Time until which data is valid.
MonitoringRef	*Monitoring-Code*	Required	Reference to a Stop Monitoring point at which visits happen. May be a SCHEDULED STOP POINT or a display identifier.
ClearDownRef	*ClearDownCode*	Optional	Identifier associated with ***MonitoredStopVisit*** for use in direct wireless communication between VEHICLE and stop display. Cleardown codes are short arbitrary identifiers suitable for radio transmission. Their scope may be transient, that is, they may be unique only to a day and sector.
MonitoredVehicleJourney.LineRef	LineCode	Required	Reference to a LINE.
MonitoredVehicleJourney.DirectionRef	DirectionCode	Required	Reference to a DIRECTION the VEHICLE is running along the LINE, for example, 'in' or 'out', 'clockwise'. Distinct from a DESTINATION.
MonitoredVehicleJourney.FramedVehicleJourneyRef.DataFrameRef		Required	Unique identifier of data frame within participant service. Used to ensure that the ***DatedVehicleJourneyRef*** is unique with the data horizon of the producer. Often the ***OperationalDayType*** is used for this purpose.
MonitoredVehicleJourney.FramedVehicleJourneyRef.Dated VehicleJourneyRef		Required	A reference to the DATED VEHICLE JOURNEY that the VEHICLE is making.
MonitoredVehicleJourney.PublishedLineName		Required	Name or Number by which the LINE is known to the public.
MonitoredVehicleJourney.DirectionName		Required	Name of the relative direction the VEHICLE is running along the LINE, for example, 'inbound' or 'outbound'.
MonitoredVehicleJourney.ExternalLineRef		Optional	Alternative identifier of LINE that an external system may associate with journey.
MonitoredVehicleJourney.OperatorRef		Optional	OPERATOR of journey.
MonitoredVehicleJo		Optional	PRODUCT CATEGORY of journey –

urney.ProductCategoryRef			classifies, for example; express, local.
MonitoredVehicleJourney.OriginName		Optional	The name of the origin of the journey; used to help identify the VEHICLE to the public.
MonitoredVehicleJourney.OriginShortName		Optional	The short name of the origin of the journey; used to help identify the VEHICLE to the public.
MonitoredVehicleJourney.Via.PlaceName		Optional Multiple	The name of a Via point of the journey, used to help identify the LINE.
MonitoredVehicleJourney.Via.ViaPriority		Optional Multiple	The priority of the Via in the same Via-structure.
MonitoredVehicleJourney.DestinationName		Optional	The name of the destination of the journey; used to help identify the VEHICLE to the public.
MonitoredVehicleJourney.DestinationShortName		Optional	The name of the destination of the journey; used to help identify the VEHICLE to the public.
MonitoredVehicleJourney.OriginAimedDepartureTime		Optional	Timetabled departure time of VEHICLE from Origin.
MonitoredVehicleJourney.DestinationAimedArrivalTime		Optional	Timetabled arrival time of VEHICLE at Destination.
MonitoredVehicleJourney.Monitored		Required	Whether there is real-time information available for journey, if not present, not known.
MonitoredVehicleJourney.InCongestion		Optional	Specifies whether the vehicle is in a traffic jam (true) or not (false).
MonitoredVehicleJourney.Occupancy	Occupancy level code	Optional	Indicates how full the vehicle is. May be one of the values ‘1’ / ‘2’ / ‘3’ / ‘4’ (passenger load level determined by LIO-System, ‘4’ means overloaded).
MonitoredVehicleJourney.BlockRef		Optional	BLOCK that VEHICLE is running.
MonitoredVehicleJourney.CourseOfJourneyRef		Optional	RUN that VEHICLE is running.
MonitoredVehicleJourney.VehicleRef		Optional	A reference to the specific VEHICLE making a journey.
MonitoredVehicleJourney.MonitoredCal		Optional	Reference to a SCHEDULED STOP POINT. Defaults to that of context i.e. that

l.StopPointRef		specified on ***MonitoredStopVisit***.
MonitoredVehicleJourney.MonitoredCall.VisitNumber	Optional	For JOURNEY PATTERNs that involve repeated visits by a VEHICLE to the same stop, the ***VisitNumber*** is used to distinguish each separate visit.
MonitoredVehicleJourney.MonitoredCall.VehicleAtStop	Optional	Whether vehicle is at stop at the current time. If absent, then the vehicle is not at the stop.
MonitoredVehicleJourney.MonitoredCall.AimedArrivalTime	Optional	Arrival time of VEHICLE in either the original or Production Timetable.
MonitoredVehicleJourney.MonitoredCall.ExpectedArrivalTime	Optional	Estimated time of arrival of VEHICLE.
MonitoredVehicleJourney.MonitoredCall.AimedDepartureTime	Optional	Departure time of VEHICLE in either the original or Production Timetable.
MonitoredVehicleJourney.MonitoredCall.ExpectedDepartureTime	Optional	Estimated time of departure of VEHICLE to show to public.
MonitoredVehicleJourney.MonitoredCall.DeparturePlatformName	Optional	Bay or platform name from which vehicle will depart.

Example of a **StopMonitoringDelivery** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <ServiceDelivery>
     <ResponseTimestamp>2013-12-30T17:44:03.000+01:00</ResponseTimestamp>
     <ProducerRef>VLOD</ProducerRef>
     <Status>true</Status>
     <StopMonitoringDelivery version="2.0">
        <ResponseTimestamp>2013-12-30T17:44:03.000+01:00</ResponseTimestamp>
        <SubscriptionRef>46</SubscriptionRef>
        <ValidUntil>2013-12-30T17:45:56.000+01:00</ValidUntil>
        <MonitoredStopVisit>
          <RecordedAtTime>2013-12-30T17:44:02.000+01:00</RecordedAtTime>
          <MonitoringRef>ANGB1B</MonitoringRef>
          <ClearDownRef>33</ClearDownRef>
          <MonitoredVehicleJourney>
             <LineRef>L1</LineRef>
             <DirectionRef>Ri B</DirectionRef>
             <FramedVehicleJourneyRef>
                <DataFrameRef>2013-12-30</DataFrameRef>
                <DatedVehicleJourneyRef>203821-00001-1</DatedVehicleJourneyRef>
             </FramedVehicleJourneyRef>
             <PublishedLineName>1</PublishedLineName>
             <DirectionName>Grubenstr.</DirectionName>
             <ExternalLineRef>1</ExternalLineRef>
             <OperatorRef>Erfurter Verkehrsbetriebe AG</OperatorRef>
             <ProductCategoryRef>Bus</ProductCategoryRef>
             <OriginName>Haupthof</OriginName>
             <Via>
                <PlaceName>1;Anger;2;Lutherkirche</PlaceName>
                <ViaPriority>1</ViaPriority>
             </Via>
             <DestinationName>Grubenstr.</DestinationName>
             <DestinationShortName>GRUS</DestinationShortName>
             <Monitored>true</Monitored>
             <InCongestion>false</InCongestion>
             <Occupancy>2</Occupancy>
             <BlockRef>100229</BlockRef>
             <CourseOfJourneyRef>59</CourseOfJourneyRef>
             <VehicleRef>469</VehicleRef>
             <MonitoredCall>
                <StopPointRef>ANGB_02</StopPointRef>
                <VisitNumber>9</VisitNumber>
                <VehicleAtStop>true</VehicleAtStop>
                <AimedArrivalTime>2013-12-30T17:38:00.000+01:00</AimedArrivalTime>
                <ExpectedArrivalTime>2013-12-
30T17:43:56.000+01:00</ExpectedArrivalTime>
                <AimedDepartureTime>2013-12-
30T17:38:00.000+01:00</AimedDepartureTime>
                <ExpectedDepartureTime>2013-12-
30T17:43:56.000+01:00</ExpectedDepartureTime>
                <DeparturePlatformName>Stg 2</DeparturePlatformName>
             </MonitoredCall>
          </MonitoredVehicleJourney>
        </MonitoredStopVisit>
     </StopMonitoringDelivery>
   </ServiceDelivery>
</Siri>
```

5.2.3.4.2 Definition of MonitoredStopVisitCancellation

Element	Type	Cardinality	Description
RecordedAtTime	*xsd:dateTime*	Required	Time at which ***MonitoredStopVisitCancellation*** was recorded.
MonitoringRef	*Monitoring-Code*	Required	Reference to a Stop Monitoring point (LOGICAL DISPLAY) at which visits happen. May be a SCHEDULED STOP POINT or a display identifier.
VisitNumber	*VisitNumberType*	Required	For JOURNEY PATTERNs that involve repeated visits by a VEHICLE to a stop, the ***VisitNumber*** is used to distinguish each

			separate visit.
LineRef	*LineCode*	Required	Reference to a LINE of journey that is being deleted.
DirectionRef	*DirectionCode*	Required	Reference to a DIRECTION of journey that is being deleted.
VehicleJourneyRef		Optional	Reference to a DATED VEHICLE JOURNEY that is being deleted.
ClearDownRef	*ClearDownCode*	Optional	Identifier associated with ***StopVisit*** for use in direct wireless communication between VEHICLE and stop display. Cleardown codes are short arbitrary identifiers suitable for radio transmission.
PublishedLineName		Optional	Name or Number by which the LINE is known to the public.
DirectionName		Optional	Reference to a DIRECTION of journey that is being deleted.
Reason		Optional	Description of the reason for failure, omitted in the case of a normal trip (departure from MonitoringRef).

Example of a **MonitoredStopVisitCancellation** message:

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <ServiceDelivery>
    <ResponseTimestamp>2013-12-30T17:39:34.000+01:00</ResponseTimestamp>
    <ProducerRef>VLOD</ProducerRef>
    <Status>true</Status>
    <StopMonitoringDelivery version="2.0">
      <ResponseTimestamp>2013-12-30T17:39:34.000+01:00</ResponseTimestamp>
      <SubscriptionRef>39</SubscriptionRef>
      <ValidUntil>2013-12-30T17:51:00.000+01:00</ValidUntil>
      <MonitoredStopVisitCancellation>
        <RecordedAtTime>2013-12-30T17:39:26.000+01:00</RecordedAtTime>
        <MonitoringRef>LUTK001</MonitoringRef>
        <VisitNumber>6</VisitNumber>
        <LineRef>L1</LineRef>
        <DirectionRef>Ri B</DirectionRef>
        <VehicleJourneyRef>
          <DataFrameRef>2013-12-30</DataFrameRef>
          <DatedVehicleJourneyRef>271501-00001-1</DatedVehicleJourneyRef>
        </VehicleJourneyRef>
        <ClearDownRef>26</ClearDownRef>
        <PublishedLineName>1</PublishedLineName>
        <DirectionName>Grubenstr.</DirectionName>
      </MonitoredStopVisitCancellation>
    </StopMonitoringDelivery>
  </ServiceDelivery>
</Siri>
```

5.2.3.4.3 Definition of StopLineNotice

Element	Type	Cardinality	Description
RecordedAtTime	*xsd:dateTime*	Required	Time at which data was recorded.
MonitoringRef	*MonitoringCOde*	Required	Reference to a Stop Monitoring point including LINE to which NOTICE applies. May be a SCHEDULED STOP POINT or a display identifier.
LineRef	*LineCode*	Required	Reference to a LINE.
DirectionRef	*DirectionCode*	Required	Identifier of DIRECTION the VEHICLE is running along the JOURNEY PATTERN, for example, "in" or "out", “clockwise”. Distinct from a DESTINATION.
PublishedLineName		Optional	Name or Number by which the LINEis known to the public. +SIRI v2.0.
LineNote		Optional	NOTICE associated with delivery.

Example of a **StopLineNotice** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.sir
i.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://ww
w.ifopt.org.uk/ifopt">
  <ServiceDelivery>
    <ResponseTimestamp>2013-12-30T17:44:03.000+01:00</ResponseTimestamp>
    <ProducerRef>VLOD</ProducerRef>
    <Status>true</Status>
    <StopMonitoringDelivery version="2.0">
      <ResponseTimestamp>2013-12-30T17:44:03.000+01:00</ResponseTimestamp>
      <SubscriptionRef>46</SubscriptionRef>
      <ValidUntil>2013-12-30T17:45:56.000+01:00</ValidUntil>
      <StopLineNotice>
        <RecordedAtTime>2013-12-30T17:44:02.000+01:00</RecordedAtTime>
        <MonitoringRef>ANGB1B</MonitoringRef>
        <LineRef>L1</LineRef>
        <DirectionRef>Ri B</DirectionRef>
        <PublishedLineName>1</PublishedLineName>
        <LineNote>Delays due to construction</LineNote>
      </StopLineNotice>
    </StopMonitoringDelivery>
  </ServiceDelivery>
```

```
</Siri>
```

5.2.3.4.4 Definition of StopLineNoticeCancellation

Element	Type	Cardinality	Description
RecordedAtTime	*xsd:dateTime*	Required	Time at which cancellation was recorded.
MonitoringRef		Required	Reference to a Stop Monitoring point.
LineRef	*LineCode*	Required	Reference to a LINE.
DirectionRef	*DestinationCode*	Required	Identifier of DIRECTION the VEHICLE is running along the JOURNEY PATTERN, for example, "in" or "out", “clockwise”. Distinct from a DESTINATION

Example of a **StopLineNoticeCancellation** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.sir
i.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://ww
w.ifopt.org.uk/ifopt">
  <ServiceDelivery>
    <ResponseTimestamp>2013-12-30T17:39:34.000+01:00</ResponseTimestamp>
    <ProducerRef>VLOD</ProducerRef>
    <Status>true</Status>
    <StopMonitoringDelivery version="2.0">
      <ResponseTimestamp>2013-12-30T17:39:34.000+01:00</ResponseTimestamp>
      <SubscriptionRef>39</SubscriptionRef>
      <ValidUntil>2013-12-30T17:51:00.000+01:00</ValidUntil>
      <StopLineNoticeCancellation>
        <RecordedAtTime>2013-12-30T17:39:26.000+01:00</RecordedAtTime>
        <MonitoringRef>LUTK001</MonitoringRef>
        <LineRef>L1</LineRef>
        <DirectionRef>Ri B</DirectionRef>
      </StopLineNoticeCancellation>
    </StopMonitoringDelivery>
  </ServiceDelivery>
</Siri>
```

6 Recovery from system failure

6.1 Recovery after restart

In the event of either a client or server restart, the client should:

1. Call the **TerminateSubscriptions** service setting parameter **ALL** = true.
2. Recreate all required subscriptions.

The client can detect a server restart by either:

- Periodically sending **CheckStatusRequest** messages to the Producer. A **CheckStatusResponse** should be returned by the Producer, including the time stamp of the last service start; or
- By monitoring for a new time stamp of the service start in HeartbeatNotification messages.

6.2 Reset after Interruption of Communication

The currently recognised failure conditions are shown in the following table:

Lost message	Lost on way to	Failure Condition	Detection	Recovery action
SubscriptionRequest	Producer	Failed to receive subscription	Subscriber received no response	Subscriber resends request again.
SubscriptionResponse	Subscriber	Failed to inform of new subscription		Subscriber resends request with same reference. Subscription is overwritten.
DataReadyNotification	Consumer	Failed to receive notification.	Producer received no reply	Producer sends notification again.
DataReadyAcknowledgement	Producer	Failed to acknowledge notification.		Resend of the request, until reply is received from the client, or a timeout.
DataSupplyRequest	Producer	Failed to receive supply request.	Consumer receives no service delivery	Consumer must assume the service delivery has been lost (worst case) and request all data again (***DataSupplyRequest*** with ***AllData***).
ServiceDelivery	Consumer	Failed to receive service delivery.		Data lost, renewed polling not possible because the server has reset the update flag of the subscription, i.e. further updates have been detected. Consumer must send a new ***DataSupplyRequest*** with ***AllData***.
TerminateSubscriptionRequest	Producer	Failed to receive subscription termination.	Subscriber received no reply	Subscriber retransmits message until it receives a reply, or there is an error message concerning an unknown Subscription Identifier or there is a timeout.
TerminateSubscriptionResponse	Consumer	Failed to receive subscription termination reply.		
CheckStatusRequest	Producer	Failed to receive status request.	Consumer received no reply	Sender retransmits up to timeout, after which it assumes Service is no longer available.
CheckStatusResponse	Consumer	Failed to receive status response.		Sender retransmits. Producer responds again up to timeout, after which it assumes Service is no longer available.

6.3 Alive Handling

6.3.1 General

There are two possibilities to detect a restart of the system:

- Status polling is used to monitor the availability of the SIRI functional services. This is done by sending a **CheckStatusRequest,** the service will respond with a **CheckStatusResponse**.
- A Consumer may detect a restart of the Producer system by monitoring for a new time stamp in **ServiceStartedTime** of **HeartbeatNotification** message.

6.3.2 Definition of CheckStatusRequest

In order to check the status of a NXTBUS service the subscriber should send a CheckStatusRequest to the [Poll Status] endpoint for the service.

Element	Type	Cardinality	Description
RequestTimestamp		Required	Time stamp of generation of request
RequestorRef		Required	Control centre code of the requesting system

Example of a **CheckStatusRequest** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.sir
i.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://ww
w.ifopt.org.uk/ifopt">
   <CheckStatusRequest>
      <RequestTimestamp>2013-12-29T14:09:41.000+01:00</RequestTimestamp>
      <RequestorRef>[Insert your API Key]</RequestorRef>
   </CheckStatusRequest>
</Siri>
```

6.3.3 Definition of CheckStatusResponse

The service will respond to a CheckStatusRequest with a CheckStatusResponse.

CheckStatusResponse	Type	Cardinality	Description
ResponseTimestamp		Required	Time stamp of creation of status information.
Status		Optional	‘true’, if the service is available, otherwise ‘false’.
DataReady		Required	If data is ready to be requested, then DataReady is set to ‘true’, otherwise to ‘false’.

ServiceStartedTime	Optional	Specifies the time of the start of the service. If the service is not available, any value may be given here.

Example of a **CheckStatusResponse** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb" xmlns="http://www.sir
i.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0" xmlns:ns3="http://ww
w.ifopt.org.uk/ifopt">
   <CheckStatusResponse>
      <ResponseTimestamp>2013-12-29T14:06:08.000+01:00</ResponseTimestamp>
      <Status>true</Status>
      <ServiceStartedTime>2013-12-29T13:35:41.000+01:00</ServiceStartedTime>
   </CheckStatusResponse>
</Siri>
```

6.3.4 HeartBeat Notification

The HeartbeatNotification message reports the availability of the SIRI Functional Service at the preconfigured heartbeat interval. A single heartbeat message is sent for each subscriber channel; if the consumer has many subscriptions, it will still get only one heartbeat.

If the System is completely unavailable there will be no heartbeat. The HeartbeatNotification message content is similar to that of CheckStatusResponse, and is also sent to the endpoint of the Data Consumer. If the Service started time is later than the creation time for the subscription, then it is likely the subscriptions are not current and that the data set of the Consumer may be incomplete.

Trapeze as data deliverer: If no HeartbeatInterval is received in SubscriptionRequest and pSiriHeartbeatNotificationInterval is not > 0s, then no HeartbeatNotification is sent.

CheckStatusResponse	Type	Cardinality	Description
ResponseTimestamp	xsd:dateTime	Required	Time stamp of creation of status information.
ProducerRef	ParticipantRefStructure	Optional	Participant reference that identifies producer of data. May be available from context.
Status	xsd:boolean	Optional	‘true’, if the service is available, otherwise ‘false’.
ServiceStartTime	xsd:dateTime	Optional	Specifies the time the service started.

Example of a **HeartbeatNotification** message:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
```

```
<Siri version="2.0" xmlns:ns2="http://www.ifopt.org.uk/acsb"
xmlns="http://www.siri.org.uk/siri" xmlns:ns4="http://datex2.eu/schema/2_0RC1/2_0"
xmlns:ns3="http://www.ifopt.org.uk/ifopt">
  <HeartbeatNotification>
    <RequestTimestamp>2013-12-29T14:09:41.000+01:00</RequestTimestamp>
    <ProducerRef>ACT</ProducerRef>
    <Status>true</Status>
  </HeartbeatNotification>
</Siri>
```