THALES

PROJECT WA063A0
ATOC DARWIN
REAL TIME TRAIN INFORMATION

OUTPUT PORTS

Push Ports

Interface Specification

P75301004 Issue 16 19th June 2014

Originator's signature & date	Approver's signature and date
D Barnard	Gareth Roberts

Thales, Ashurst Drive, Bird Hall Lane, Cheadle Heath, Stockport, Cheshire SK3 0XB, UK
Tel: +44 (0)161 491 4001•Fax: +44 (0)161 741 3704•ukinfo@thalesgroup.com•www.thalesgroup.com

DISTRIBUTION

1 Project Master File

ISSUE RECORD

Issue	Date	Purpose
1 Draft A	04/04/2003	First draft issue
1 Draft B	14/04/2003	Darren Barnard comments
1 Draft C	20/06/2003	Inclusion of Kizoom and Bob Machon comments
1 Draft D	07/07/2003	Further Kizoom comments
1 Draft E	21/07/2003	Further Bob Machon comments
1 Draft G	17/12/2003	Introduction of CCR10 – Timetable via push ports
1 Draft H		
1 Draft I	08/03/2004	Introduction of CCR14 – Overdue/uncertainty
1 Draft J	18/03/2004	Modifications to CCR10 updates
1 Draft K	25/05/2004	Bob Machon's comments on Issue 1i addressed. Document made more 'stand-alone' by including details taken from the RTTI FDS P75301001.
1 Draft L	4/06/04	Alan Gawthorpe's review comments incorporated.
1 Draft M	13/07/04	Flags odet and nr introduced to examples. Table Suppression field - <Suspend/> changed to <Suppress/> Review comments from Alan Gawthorpe incorporated.
1 Draft N	20/08/2004	
1 Draft O	12/01/2005	Made to match more closely the format of Enquiry Ports. Added a namespace to the schema for versioning. Added platform numbers and message categories for future use.
1 Draft P	28/02/2005	Updates to message categories. Changed PPort to Pport to match Enquiry Ports style.
1 Draft Q	02/06/2005	Incorporated comments from Bob Machon.
1 Draft R	09/06/2005	Added 'id' attribute to operator messages.
1 Draft S	15/06/2005	Added document changes from version P suggested by customer.
Issue 1	15/06/2005	Issued.
Issue 2a	07/09/2005	Initial version two of the schema. Includes suggestons from the workshops. Removed 'nr' attribute – actually a responsibility of the client. Changed examples to a fixed width font. Added information on FTP snapshots.
Issue 2b	15/09/2005	Incorporated comments from Mark Shields. Change bars for 2a left in place.
Issue 2c	27/10/2005	Changed instructions for FTP.
Issue 2d	16/11/2005	Added support for Q Trains. Changed 'id' to 'trainId' where appropriate to match Enquiry Ports.

Issue 2	25/11/2005	Issued.
Issue 3a	16/02/2006	ECN303 Add "expired" (bool) attribute to "Overdue" element.
Issue 3	17/02/06	Issued.
Issue 4a	17/05/06	CPR245 Update the schema so that the Schedule.IP ptd attribute is optional. Incorporate CCR55 Bus and Ferry Information. Update schema to version 3.
Issue 4	19/05/06	Issued.
Issue 5a	20/06/06	CCR55 Output Train Category
Issue 5	23/06/06	Issued.
Issue 6a	17/09/07	CCR62 Add uncertain indicators to Actual and Forecast info.
Issue 6	18/09/07	Issued.
Issue 7a	12/12/07	Add ‘deleted’ flag to schedules (PP schema v5)
Issue 7	12/12/07	Issued.
Issue 8	08/10/09	CCR11v5 Ad-hoc Alerting Added “delayed” attribute to “TS” element
Issue 9a	01/12/09	Toc Code to be sent for Ad-hoc Alerts (NRERTTITEST-1570)
Issue 9b	07/06/10	Update to support multiple schema versions. (D Goodwin) Reorganisation & extra clarification (D Barnard).
Issue 9	10/06/10	Issued.
Issue 10	08/07/10	Updates following comments from Amey and NRE (D Barnard).
Issue 11	12/11/10	Minor update to examples (DB) Updates for PP schema v8 (DB)
Issue 12	07/01/11	Support for filtered ports. V2 of the setup schema. (DB)
Issue 13	05/08/11	Added passing points and working times to schema (DG & DB)
Issue 14	06/09/11	Update timetable schema to make activity codes consistent with data schema.
Issue 15	11/03/13	Fix namespace case errors. Initial DCIS support for code drop 1. (DB)
Issue 16a	19/06/14	Added originSource and working forecast time for public calling points to schema (v12). Minor clarifications.

CONTENTS

OUTPUT PORTS
PUSH PORTS
INTERFACE SPECIFICATION

1. Introduction

Push Ports are provided to meet the needs of external systems that deal with high volumes of enquires and that require rapid access to Darwin (RTTI) information. Darwin provides a "push" service of information in real-time that allows the client system to hold a copy of the Darwin database.

This document addresses the Interface Specification for the retrieval of information via the Darwin (RTTI) Push Ports Interface.

Darwin makes available via the Push Port creation of, and changes to, train schedule records, together with train running predictions made by Darwin. Note that predictions and changes to schedules are relative to the original schedule as created in Darwin, usually sourced from ITPS. In order to correctly interpret Darwin data, clients must also have access to the ITPS schedule data.

Darwin also supports the download by FTP of a complete XML-format timetable for those Clients who do not have access to their own ITPS-generated timetable.

Darwin holds a minimum of 48 hours worth of data with the database being rebuilt overnight. The Client's database must mirror this period if they wish to process all Push Port messages.

Thales will allocate a dedicated TCP/IP port and FTP account to each Client that wishes to use the Push Port Interface.

Darwin also provides a separate interface (DCIS web services) to allow clients to make updates to Darwin data. This Push Port interface also provides some features to allow these DCIS clients to provide a full end-to-end solution, from requesting an update to the subsequent Push Port updates that it generates. Features added to this Push Port interface to support DCIS clients can be used by non-DCIS clients, though such features may not always be available to all clients, subject to configuration or other reasons.

DCIS clients will also be allocated a unique identifier that the Darwin system uses to identify them in its configuration. This "update source" identifier is provided in certain Push Port messages to allow a DCIS client to identify updates that it has itself caused.

2. References

1. Common Interface File, End User Specification, Issue 28, Nov. 2012. Issued by Network Rail.

3. Communication

Physical network connection between the Client network and Darwin is not specified here, but will typically be either a dedicated Leased WAN connection installed and maintained by Thales, or a pair of VPN connections operating over the Internet. The difference between these two options related to Client connectivity mainly involves resilience features in the event of a failure within the Darwin System.

When use is made of the dedicated WAN connection, failures within the Darwin system are transparent (at the communications level) to the Client system (other than a brief loss of connectivity). The WAN connection will be automatically re-configured to connect to an active Data Centre.

If VPN connections are used, there must be a separate VPN to each of the Darwin Data Centres and it is the responsibility of the Client to implement any logic and network routing to connect to any active Data Centre.

Security for the communication link is provided by use of the Leased WAN, or over the Internet by VPN encryption. The Darwin Firewall will only allow connection from known IP addresses to a dedicated port. Push Port information is not deemed to be highly sensitive, so these measures are considered adequate.

3.1 Connection

Push Port communication is carried out over a persistent TCP/IP socket connection, initiated by the Client. Thales allocates each Push Port Client a dedicated TCP/IP port. Only one simultaneous connection to this port is permitted at any one time.

Client systems communicate with the Darwin System by connecting to a Darwin Push Port server. The Push Port Client application must specify the following in order to make a connection to Darwin:

- IP Address of the Darwin host.
- Port Number allocated to the Push Port Client.

3.2 FTP

FTP is used by clients to retrieve timetable data and optionally snapshot data. Each client that requires FTP access will be provided with the IP address of the FTP server, a username and password. The account will be read-only and clients cannot write or delete files.

3.3 Availability

The Push Port service is available continuously, 24x365. Data is provided as soon as it is updated, other than when Darwin is re-building its timetable, when it may be queued for a short time. Darwin typically re-builds its timetable at 02:30 GMT each day, though this may possibly occur at other times due to operational reasons.

4. Protocol

All data sent from Darwin or a Push Port Client will be formatted as XML messages. To make it easier to write code to process these messages, each individual XML message shall be preceded by an STX character (ASCII code 0x02) and followed by an ETX character (ASCII code 0x03).

To reiterate this:

All requests to Darwin *must* have a STX character preceding them and an ETX character following them.

All responses from Darwin *will* have a STX character preceding them and an ETX character following them.

A client that wishes to process a response from Darwin should read all data between an STX/ETX pair then load the message contained between these characters as an XML document.

4.1 Multiple Version Support

In order to allow new Push Port features to be introduced without forcing all existing clients to update all at once, the Darwin Push Port service supports multiple simultaneous versions. In order to provide this multiple version capability, the setup of a Push Port Client connection is split into two separate phases, a version-independent Setup Phase and a version-dependent Data Phase. Each of these phases is defined in separate XML schemas, the current versions of which can be found in section 7 of this document.

Figure 1 shows a flowchart of the expected actions a Client will perform when setting up a Push Port connection. The yellow blocks represent the Setup Phase and the blue blocks represent the Data Phase.

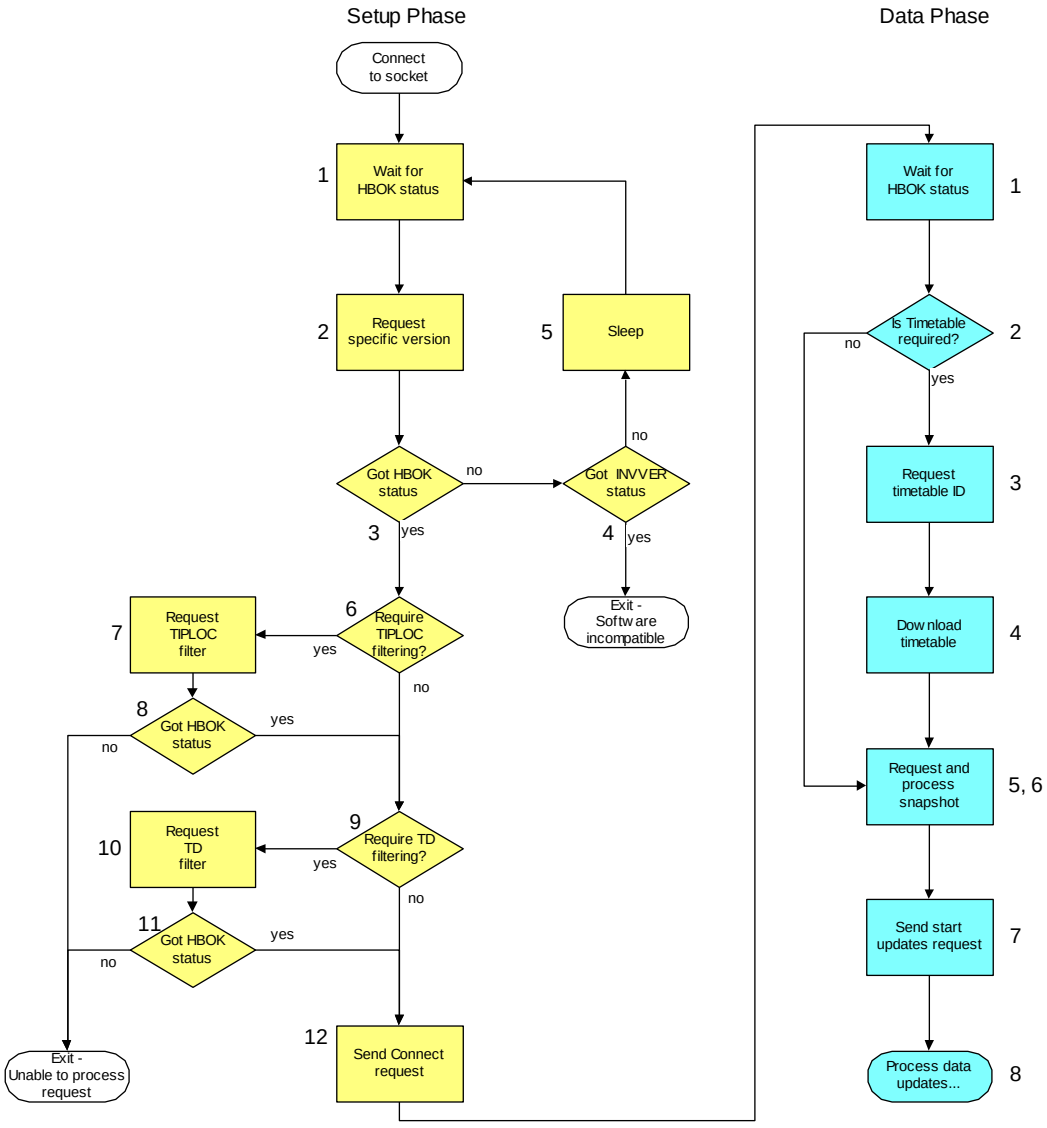

Figure 1 - Client Connection Flowchart

4.2 Setup Phase

1. As can be seen in Figure 1, after a client connects to the Push Port socket, it must wait for the server to send a *Status* message with a status of "*HBOK*" before proceeding. This allows the Push Port server to determine that it is in the correct state before accepting new connections.

```
<PPStatus code="HBOK">System is available</PPStatus>
```

2. The next step that a client is expected to perform is to request the specific versions of the data schemas that the client supports. There are three data schemas, the real-time updates schema, the timetable schema and the reference data schema. The client requests each of these individually by specifying the XML namespace of the schema. Note that if a client does not wish to download a timetable a valid ttversion and ttrefversion attribute must still be supplied. Also note that the versions given in

the examples below are not necessarily the most current or most suitable for any particular client. A client must obtain the correct values from the exact version of the schema files that they wish to use.

```
<PPReqVersion version="http://www.thalesgroup.com/rtti/PushPort/v7"
ttversion="http://www.thalesgroup.com/rtti/XmlTimetable/v5/rttiCTTSchema.xsd"
ttrefversion="http://www.thales-is.com/rtti/XmlTimetable/v1/rttiCTTReferenceSchema.xsd" />
```

3. On receipt of the version request, the server validates that it supports each of the requested schemas and returns a "*HBOK*" status if so. Each of the requested schema versions is associated with the client's connection.

4. If one or more of the requested versions is unknown then an "*INVVER*" status is returned. On receiving this status, a client is informed that it is incompatible with the Push Port server and should immediately exit.

5. If a client receives any other status that it cannot otherwise handle, it should sleep for a period (to stop busy looping) and try to make the connection again later.

```
<PPStatus code="INVVER">Invalid version requested</PPStatus>
```

6. If a client requires a *filtered* push port (see sec. 4.2.2), proceed to step 7, otherwise continue at step 9.

7. The client now sends a filter request with a set of filter TIPLOC codes

```
<FilterTiplocs>
    <tiploc>MNCRPIC</tiploc>
    <tiploc>STKP</tiploc>
    <tiploc>GATLEY</tiploc>
</FilterTiplocs>
```

8. In order to continue, the client must receive a "*HBOK*" status response. If any other response is received, the suggested action is for the client to close the socket and exit. If the client receives an "*INVREQ*" response to a filter request, this is a sign of a configuration error (most likely that one or more of the requested TIPLOC codes is not recognised by the Push Port server).

9. If a client requires a *TD filtered* push port (see sec. 4.2.3), proceed to step 10, otherwise continue at step 12.

10. The client now sends a TD filter request with a set of filter TD area codes

```
<RequestTD>
    <td>MS</td>
    <td>E1</td>
    <td>HN</td>
    <td>MP</td>
</RequestTD>
```

11. In order to continue, the client must receive a "*HBOK*" status response. If any other response is received, the suggested action is for the client to close the socket and exit.

12. Finally, the client will send a *Connect* message, which instructs the server to switch to the requested real-time update Data schema and operate as specified later in the Data Phase. Note that the server will not send a "*HBOK*" status response to a *Connect* message in the Setup Phase schema. If the connection is successful, the next message will be a status message in the requested Data schema.

```
<PPConnect />
```

There is the possibility that a client sends a *Connect* request before sending a *Version* request (or a client may ignore an "*INVVER*" response and try to *Connect* anyway). In this case, the server will respond with an "*INVREQ*" status then close the socket connection, thus indicating to the client that the correct protocol has not been followed.

4.2.1 Setup Schema Extensibility

The Setup schema in Figure 2 is designed to be extensible, so that different versions of clients can use the schema without breaking if the schema is updated. To achieve this, the schema is defined with just two simple element definitions, for request (incoming to Push Port server) messages and response (outgoing to clients) messages. These elements are declared with the known set of messages that can be sent as a request or response, but each also includes an <xs:any/> declaration, to indicate that any other element (from any other namespace) can also be present.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://thalesgroup.com/RTTI/PushPortSetup/root_1"
xmlns:pp1="http://thalesgroup.com/RTTI/PushPortStatus/root_1"
xmlns:pp2="http://thalesgroup.com/RTTI/PushPortFilter/root_1"
xmlns:pp3="http://thalesgroup.com/RTTI/PushPortRequestTD/root_1" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://thalesgroup.com/RTTI/PushPortSetup/root_1" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="2">
    <xs:import namespace="http://thalesgroup.com/RTTI/PushPortStatus/root_1"
schemaLocation="rttiPPTStatusV1.xsd"/>
    <xs:import namespace="http://thalesgroup.com/RTTI/PushPortFilter/root_1" schemaLocation="rttiPPTFilterV1.xsd"/>
    <xs:import namespace="http://thalesgroup.com/RTTI/PushPortRequestTD/root_1"
schemaLocation="rttiPPTRequestTDV1.xsd"/>
    <xs:element name="PPSetupReq">
        <xs:annotation>
            <xs:documentation>Definition of request messages from clients</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:choice>
                <xs:element ref="pp1:PPReqVersion"/>
                <xs:element ref="pp1:PPConnect"/>
                <xs:element ref="pp2:FilterTiplocs"/>
                <xs:element ref="pp3:RequestTD"/>
                <xs:any processContents="lax"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="PPSetupResp">
        <xs:annotation>
            <xs:documentation>Definition of response messages to clients</xs:documentation>
        </xs:annotation>
```

```
        <xs:complexType>
            <xs:choice>
                <xs:element ref="pp1:PPStatus"/>
                <xs:any processContents="lax"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Figure 2 – rttiPPTSetupV3.xsd Schema

A client is required to **ignore** any response elements that it receives that are not defined in the version of the schema that it uses. The server will always have the most recent version of the schema (by definition) but similarly will ignore unrecognised request elements.

When the Setup schema is required to be extended, new request or response messages will be added by importing a new schema definition containing the message definitions and including the appropriate element references in the <PPSetupReq> or <PPSetupResp> elements. Note that the schema targetNamespace will *not* be updated in this case, though the version attribute may be.

The only exception to the above rule is for the <PPStatus> element, which is required to be understood by all versions of clients. To allow the capability that this element can be updated, it has been defined with the <xs:anyAttribute /> declaration. Clients are required to **ignore** any attributes of this element that they do not recognise, thus allowing future extensions to be added.

4.2.2 Filtered Push Ports

Beginning with version 2 of the Setup schema, a client may optionally request a *filtered* push port. Requesting a *filtered* port affects just the Setup phase and may be used with any supported Data phase schema version.

To request filtering, a client issues a <FilterTiplocs> request, after requesting the required schema versions and before making a <PPConnect /> request. If a <FilterTiplocs> request is not made before a <PPConnect /> request then the push port is not filtered.

Supplied with the <FilterTiplocs> request is a list of one or more <tiploc> elements. These elements define the stations for which filtered data is required. If a station has more than one TIPLOC code associated with it, then all TIPLOC values must be supplied in order for all data to be received. TIPLOC codes for multiple stations may be supplied, if required.

Normally, a client may only filter on calling points (i.e. locations with a passenger activity). Passing services, where a train does not stop at a filtered location, will not be returned. However, it is possible for Thales to enable configuration to allow passing trains to be sent. If a client requires such configuration, then a request should be submitted to NRE.

When a port is filtered, the server will only send data to a client that is relevant to one or more of the filter locations, or has been explicitly activated by the client associated with the port using the DCIS Web Service interface. Thus, updates for a service will be sent if that service calls at one of the filter locations (or is associated with a service that calls at a filter location). Station messages (<OW> elements) will only be sent if they apply to a

filter location. Note that this applies equally to both “normal” update messages and snapshot messages. However, also note that the download-able Timetable files (see sec. 5) will not be filtered and will always contain information for all services and locations.

Once a filtered port sends data for a service, it will continue to send updates to that service even if the service no longer meets the filter criteria. Thus, if a service is edited to call at a filter location, it will be sent to the client. If that service is subsequently edited to remove that filter location, the updates to the service will still be sent to the client. Once a service is associated with a filtered port, it will not be un-associated until the service is removed from the Darwin timetable.

4.2.3 Request TD Filtering

Beginning with version 3 of the Setup schema, a client may optionally request to receive certain data related to Train Describer (TD) information. The exact data that is received is documented later, in section 4.3. However, in order to receive any of this data, a client must explicitly request those TD area codes for which they require data.

To request filtering, a client issues a <RequestTD> request, after requesting the required schema versions and before making a <PPConnect /> request. If a <RequestTD> request is not made before a <PPConnect /> request then no TD-related data will be sent. The order in which <FilterTiplocs> and <RequestTD> requests are made is not significant, and either, none or both may be sent in any setup request.

A <RequestTD> request includes one or more <td> elements, which contain a single two-character TD area identifier. Darwin will not validate that a particular area identifier is valid in any way, but will only send TD data that exactly matches the supplied value.

4.3 Data Phase

> Note that from version 11, the data schema has been broken into several separate files. This has been done for ease of management and to localise the impact of future changes. From a usage perspective little has changed, as there is a single root schema that includes the other files.

There are slight variations on the protocol dependent on whether the client uses train schedules sourced from Darwin and whether the client requests snapshot data via ftp or back over the Push Ports connection.

As indicated in Figure 1, Clients must follow these steps when communicating with Push Ports Server in the Data Phase:

1. The client waits for an HBOK status message.
2. If the client does not require the Darwin sourced timetable, go to step 5.
3. The client requests the timetable ID.
4. The client downloads the timetable from the Push Ports FTP server if the ID is different to the timetable the client already has, using the supplied file names.
5. The client requests a snapshot; either back over the Push Ports connection or made available on the FTP server (see section 6.1.2.3).

6. The client processes the snapshot.

7. The client sends a start update request (see section 6.1.2.4). Push Ports will start holding updates for the client from the moment the snapshot request is received. These buffered updates are sent as soon as the client sends the start update request.

8. The client processes updates until a status message with either HBFAIL or HBINIT is received, the client requests the cessation of updates or the TCP/IP connection fails.

9. If the TCP/IP connection has not failed, go to step 1.

Darwin will respond to Client requests with either the information requested or a Darwin status message to indicate the reason for the failure to meet the request. Darwin will also broadcast heartbeat messages.

A list of heartbeat and status messages is provided in section 4.3.1.

Note: A database rebuild occurs on a nightly basis. Currently, this is configured to occur at 2:30 am GMT, although this is subject to change and should not be relied upon. During the rebuild, Darwin will not issue any update messages and the heartbeat will indicate that the database is being re-initialised (HBINIT). Once the database rebuild has completed, Darwin will notify the Client that the database is available via the heartbeat message (HBOK). At this point the client should resume the above sequence at Step 1.

4.3.1 Darwin Status Messages

Status messages are sent in response to some client requests. A status message with the current state of the interface (HBOK, HBPENDING, HBINIT or HBFAIL) is also sent every 60 seconds if no other data is sent to the client in that time. This message is also sent immediately if the state of the interface has changed.

The status messages that may be received by the client are:

Code	Type	Text	Description
HBOK	Heartbeat; sent periodically.	System is available	Darwin is running and able to accept requests for data.
HBINIT	Heartbeat; sent periodically.	System is initialising	Darwin is running but is initialising its timetable. Clients should wait until a HBOK message is received.
HBFAIL	Heartbeat; sent periodically.	System is unavailable	Darwin is shutdown (the push port handler is a separate process from the core Darwin process).
HBPENDING	Heartbeat; sent periodically.	System is failing over and data is delayed	Darwin is operating, but part of the system is currently in failover mode. Data may be queued for a short period. Clients may remain connected and data will be delivered when available. This status is only returned in data schema version 11 and later.
XMLBUSY	Error; a problem was detected.	System is responding to a previous request	Darwin is busy responding to a previous request by the Client. Client should wait for the 'HBOK' code.

XMLBADIN	Error; a problem was detected.	Failed to validate request	The request XML did not match the schema.
SOFTERROR	Error; a problem was detected.	Something went wrong processing the request	An error occurred when processing the request. These errors should be reported to Thales for investigation.
XMLBADOUT	Error; a problem was detected.	Failed to generate a valid response.	Push Ports generated a response that was invalid against the schema. These errors should be reported to Thales for investigation.
TOOLONG	Error; a problem was detected.	Too much data received	Too much data has been received without a valid Push Port XML end tag.
INVREQ	Error; a problem was detected.	Dependent on circumstance.	An invalid request was made.
INVVER	Error; a problem was detected	An unsupported schema version has been requested.	A request version message was sent with schema versions not supported on the push port server.
QOVERFLOW	Error; the output queue has overflowed	Output queue overflow	The client is taking too long to process messages and the Darwin output queue has overflowed. Data will have been lost.

The general format of Darwin Data Phase status messages is as follows:

```
<?xml version="1.0"?>
<Pport  …>
  <FailureResp code="HBOK">
    System is available
  </FailureResp>
</Pport>
```

From version 11 of the data schema, DCIS clients can request (via the DCIS web service interface) a heartbeat operation, to verify full end-to-end operation. When responding to a heartbeat request, a <FailureResp> message will include optional "requestSource" and "requestID" attributes. These attributes allow a client to detect that this heartbeat message was generated as the result of the DCIS web service request made by that client, verifying end-to-end operation. Only the client that requested the heartbeat, as determined by the "requestSource" attribute, will receive the message. The "requestID" attribute is an optional value provided by the DCIS client with their heartbeat request.

The status returned for a heartbeat will reflect the current state of the system, as will be returned in the next regular status message (assuming the state does not change in the meantime). During some internal Darwin failover scenarios, heartbeat messages may be lost, even though the regular status messages appear to indicate that the system is available.

4.3.2 Client Requests

A Client registered for using the Push Ports interface is able to send the following requests to the Darwin System:

- Timetable ID Query
- Request Snapshot

- Start/Stop Updates

4.3.2.1 Timetable ID Query

A client may request the current timetable ID to check if the Darwin timetable has been re-built during a period of disconnection. Clients that remain connected will be informed of timetable re-builds via a HBINIT status message, though the timetable ID should still be checked when the HBINIT turns into HBOK, since the ID may not always change for every HBINIT.

4.3.2.2 Request Snapshot

When a client is ready to start receiving real-time updates, after downloading a timetable if necessary, then a snapshot request must be made to synchronise with the current state of Darwin's data.

Snapshots summarise the information already held by Darwin. There are two types of snapshot available. As a part of a **Standard Snapshot**, Darwin provides information for all *activated* train journeys in the Darwin database (see section 4.3.4.1), even if the service has not been modified.

As part of a **Full Snapshot,** Darwin provides the same information as the Standard Snapshot but for all services that are *activated*, or have been modified from the base ITPS planned timetable in some way. Timetable services that have not been modified and are not active shall not be sent (though are available in a Timetable file (see sec. 5)).

As soon as a snapshot request is made, Darwin will start queuing subsequent updates so that they can be delivered when the snapshot is complete. This queue is of limited size, so to avoid it overflowing, the snapshot must be processed without excessive delay. If the queue does overflow, updates will be lost. In this case, a one-off QOVERFLOW status message will be sent at the next opportunity after completion of the snapshot. Note that handling this error by disconnecting and reconnecting is unlikely to achieve anything. The next snapshot is likely to also have the same effect. Clients should read messages in a separate thread to the processing of them and queue them internally if they cannot process them immediately.

When the snapshot has been fully received and processed, real-time updates will only resume once a "Start Updates" request has been issued.

4.3.2.3 Start/Stop Updates

When a client wishes to start processing real-time updates following a snapshot, the "Start Updates" request must be made. This will flush the queue of updates that were collected while the snapshot was processed and allow further updates to be sent as they occur. However, if the client is unable to receive updates as fast as they are generated, the updates will be added to the queue mentioned above. If this state persists, eventually this queue may overflow and updates will be lost. In this case, a one-off QOVERFLOW status message will be sent at the next opportunity.

A client may optionally issue a "Stop Updates" request at any time after "Start Updates" has been requested. This will stop any further real-time updates from being sent and will cause the port to return to the state where it is expecting a snapshot request. Real-time

updates will not be queued until a snapshot request has been made. Regular status messages will be issued in the normal way.

4.3.3 Snapshots and Updates

The information provided to a Push Port Client (either as a snapshot or an update) comprises of:

- Schedule information (see section 4.3.4)
- Association information (see section 4.3.5)
- Actual and forecast information (see section 4.3.6)
- Train order information (see section 4.3.7)
- Status of train table suppression and station messages (see section 4.3.8)
- Train alert information (see section 4.3.10)
- Tracking ID Corrections (see section 4.3.11)
- Alarm information (see section 4.3.12)

DCIS clients can originate updates to Darwin data (via the DCIS web service interface). From version 11 of the data schema, when the Push Port generates an update response (<uR>) message directly resulting from that update, it will include values for optional "requestSource" and "requestID" attributes. These attributes allow a client to detect that this update message was generated as the result of the DCIS web service request made by that client. The "requestID" attribute is an optional value provided by the DCIS client with the original request.

Additionally, from version 12, an update response may also contain an "updateOrigin" attribute. This indicates the origin source type of the update, for example "CIS", "TD", "Tyrell", or others. Updates from origins other than "CIS" may or may not have an associated "requestSource" attribute, but if this attribute is present, it is guaranteed not have a value that duplicates a "requestSource" allocated to a CIS.

Snapshot responses do not include these "requestSource" and "requestID" attributes, as snapshots are always specific to an individual Push Port client and are not generated from a DCIS web service request.

When sending updates, Darwin will send schedule changes first followed by updates to the forecast/actual times.

In some circumstances, a DCIS web service request may be accepted by Darwin, but will generate no actual change to the data. In order for a DCIS client to get positive feedback that their request has completed, Darwin will send an empty update response (<uR>) message to confirm the update. This empty message will only be sent to the port(s) associated with originating DCIS client.

4.3.4 Schedule Information (the 'Schedule' element)

As a part of an **Update**, Darwin provides schedules when a new schedule or schedule change becomes known to Darwin, or the schedule is activated.

A schedule covers the complete journey of one train.

Schedule changes and new schedules are always transmitted in full. Schedule changes replace previous versions of the schedule.

Note: Journeys are primarily identified by RID (Darwin generated ID). The RID is unique in the Darwin database whereas the UID is not.

A schedule comprises of an origin, zero-to-many intermediate points and a destination. Intermediate points may be calling points (where the service stops for passengers) or (from v9 onwards) passing points. From v11 onwards, operational calling points may be supplied, where a train calls at a location for operational reasons, not applicable to passengers. Cancelled locations are also included in the schedule. If an origin/destination is cancelled, then it will still be identified as an origin/destination but there will be an additional point that represents the live origin/destination. It is possible for cancelled locations to appear before the live origin and/or after the live destination. Locations will be listed in order from origin to destination. This will normally be in chronological order by working scheduled times, but prior to version 11 this is not guaranteed in all circumstances, particularly if a service is re-routed from its booked path. Note that Darwin makes no guarantee that the order of locations implies any valid route on the physical rail network. Neither is there any indication whether cancelled locations will be passed through by the running train, or not.

From version 11 of the data schema, when a service is re-routed to a new path, which then re-joins the original path, the location at which the service re-joins may have an "rdelay" attribute. This attribute provides a delay value that is implied by the change to the service's route. Darwin will add this value to the forecast lateness of the service at the previous schedule location when calculating the expected lateness of arrival at this location. A client that is expecting scheduled times to chronologically increase will need to take this value into account, since the scheduled times may jump back when the service joins its original route. Adding the "rdelay" value to the scheduled times at the location where it is defined, plus later locations, will maintain chronological order.

In determining the chronological order of locations, where cancelled locations imply an overlapping range of scheduled times, Darwin will sequence the locations based in the working arrival, pass or departure times (as applicable), in that order. For example, given a cancelled location "A" with sta="10:00:00" and std="10:05:00" and another location "B" with stp="10:03:30", then location "A" will be ordered before location "B".

For clients using a data schema prior to version 11, when a service is re-routed and has a "rdelay" value applied, Darwin will not publish this value. Thus, it is possible that the scheduled times will not be in chronological order at the point where the service re-joins its original route. A client needs to be able to cope with these discontinuities, but in order that correct behaviour in all cases can be achieved, it is recommended that clients update to the latest schema version.

Since there is no absolute guarantee that live times are in chronological order, a client must correctly handle the case where a time goes backwards, or just appears to do so

because it has crossed a midnight boundary. Darwin uses the following rules to handle these cases:

Difference Between Two Times	Interpret As
Less than –6 hours	Crossed midnight
Between –6 and zero hours	Back in time
Between zero and +18 hours	Normal increasing time
Greater than +18 hours	Back in time and crossed midnight

From version 9 of the data schema onwards, there are two sets of scheduled times for schedule locations: Working scheduled times and Public scheduled times. All locations have Working times, but only locations that have a passenger activity normally have Public times. In general, only Public times should be used when displaying schedules. Note that Working and Public times may, and commonly do, differ from each other. Note that in the version 9 schema, the working times were defined with the wrong data type, which truncated the number of seconds in the time. This has been corrected in later versions.

Note that also from version 9, the definition of schedule activity codes has changed slightly. Previously, they consisted of 12 character fixed length data, reflecting the way they are provided from CIF. From version 9 onwards, activities have been re-defined to be 6x2 character data, with empty activities consisting of two spaces (" ") removed. Thus a typical activity of "T " has changed to "T ".

It is possible that Working and Public times will be provided at locations where the activity codes indicate that they are not valid. For example, if a train is terminated short, it will have one of its calling points modified to have a “TF” activity, to reflect the fact it is now the destination. However, the scheduled departure times will still be published, to allow clients to show the planned departure has now been cancelled.

From version 11 of the data schema, to allow clients to detect when activities have been modified, planned activity codes are provided when they differ from the current activities. Clients of prior versions do not have planned activity codes supplied, and consequently cannot detect when they are changed without remembering them across changes, as they happen.

It is possible for schedules to be marked with a ‘Deleted’ flag. This is used to indicate services that exist in the Darwin database, but have been manually removed from public display. If a schedule is received with the ‘deleted’ attribute set to true then the client should act as if the schedule does not exist. Note that the ‘deleted’ flag can be set or cleared at any time, including for services that are in progress.

Individual schedule locations may also be deleted within Darwin. Clients of all versions will not receive locations marked as deleted, or any indication of their deletion, other than their absence from the schedule.

Schedules and schedule changes will include false destinations and cancellation reasons if available. False destinations apply on a per location basis.

Prior to version 11 of the data schema, cancellation reasons are provided per schedule location. From version 11 of the data schema, only a single cancellation reason is provided for the whole service.

Prior to version 11 of the data schema, a schedule location may also include a platform number, so that scheduled platforms may be provided when schedules are created. Note however, that the normal way of receiving an update to platform numbers is by a *TS* update. From version 11 of the data schema, platforms are not provided with a schedule location, and will only be provided in *TS* updates. If required, a TS update will be sent when a schedule is created to communicate platform (and other) information.

Darwin will also provide Bus and Ferry information. The transport service type will be identified by a ***status*** attribute within a schedule. The value used by the ***status*** attribute is defined in reference 1, though Darwin does not support Freight and Trip values. A service is treated as a train, bus or ferry solely based on the status value.

The Train Category is identified by the ***trainCat*** attribute within a schedule. The value used by the ***trainCat*** attribute is defined in reference 1.

From version 11 of the data schema, Darwin will support Empty Coaching Stock (ECS) trains in its data. ECS trains are identified by the value of the Train Category. Darwin will interpret a train with a category set to one of the following values as a passenger train: “OL”, “OO”, “OW”, “XC”, “XD”, “XI”, “XR”, “XX”, “XZ”. All other values of Train Category for a train shall be interpreted as an ECS (non-passenger) train. To aid clients in interpreting the Train Category values, an ‘isPassengerService’ attribute will be set on a schedule to true when one of the previous list of passenger category values is set. The ‘isPassengerService’ value will be set to false for any other value of category.

From version 11 of the data schema, a schedule may be marked with a boolean attribute to indicate that it is a charter service. Clients of prior versions will still receive the schedule, but there will be no indication that it is a charter service. Instead, the service will be marked as deleted.

4.3.4.1 Schedule Activation

From the release of version 11 of the data schema, for all supported schema versions (including those prior to version 11), Darwin will send a *Schedule* update to clients when the service is *activated*. Activation may occur at any time, but will typically be at a configurable window before scheduled departure from origin. Activation may also occur, if it has not already happened, prior to any update being applied to a service. Thus, each service will always send a *Schedule* message and a *Schedule* message will always be the first message received by a client.

If a DCIS web service client requests to activate a schedule and that schedule is already active, then the subsequent Push Port *Schedule* update will only be sent to the port(s) associated with originating DCIS client.

Where an activated service has Join, Divide or Link associations, these associated services will also be activated at the same time. Darwin will publish all associated schedules and their *association* elements in the same message as the schedule that was activated. *Schedule* elements will appear before the *association* elements that reference them in the published message. Darwin will also publish any Next associations containing the activated

service, but will not activate or publish (in the same message) the other schedule referenced by a Next association.

A schedule message is also sent for Q Trains that become active, though this will only happen when Darwin has positive confirmation that the service will run. See section 5.2.1 for details on these journeys.

At a time after which Darwin no longer anticipates further updates to a service, it may be *deactivated*. Services that become deactivated will be notified by a *deactivated* message, which simply identifies the RID of the deactivated service. Deactivated services should not be published to the public.

If Darwin receives a subsequent update for a deactivated service, it may reactivate it by sending another *Schedule* message. The reactivated schedule may still meet the criteria for deactivation, but any subsequent *deactivated* message is guaranteed not to be sent for at least 60 seconds.

Since Standard snapshots only include activated schedules, schedules that are deactivated will not appear in Standard snapshots. However, if the service has been modified in any way, a Full snapshot may include a schedule for a deactivated service, but in this case it will be marked with an `isActive="false"` attribute.

Prior to version 11 of the data schema, clients will not receive *deactivated* messages or snapshot schedules with an `isActive="false"` attribute. Thus, these clients will not be informed of deactivation.

4.3.5 Association Information (the 'association' element)

From version 11 of the data schema, association information has been provided as *association* elements separate from *schedule* elements. Associations consist of:

- a category, which can be Join, Divide, Link or Next
- a TIPLOC location where the association occurs
- a cancellation flag, to indicate that the association is no longer happening
- a deletion flag, to indicate that the association no longer exists
- a "main" service identifier
- scheduled times for the association location on the main service (for use on circular routes where the location may appear more than once)
- an "associated" service identifier
- scheduled times for the association location on the associated service (for use on circular routes where the location may appear more than once)

For Join and Divide associations, the main service is that which continues through the TIPLOC location. The associated service is that which terminates or starts at the location. For Link associations, the main service is that from which passengers transfer at the TIPLOC location, to the associated service. For Next associations, the main service is the one which provides the rolling stock to form the associated service as its next working.

Multiple associations may exist for a service at a single location, but only one association of the same category may exist between the same two services at a location, whether cancelled, deleted, or not. Only one non-deleted association (of any category) may exist between the same two services at a location.

If an association is marked as deleted, the association may not meet the usual rules for valid (non-deleted) associations. For example, a deleted association may refer to a TIPLOC location that does not exist in either of the associated schedules.

4.3.5.1 Link Associations

Link Associations are provided to allow two services to be linked together to provide a single "combined view" of a service to the public. These are typically used in scenarios where a Bus Replacement service is linked to a train running only part of its schedule, to make it appear that it is still running for most or all of its original calling pattern. However, services may be linked in other circumstances, as deemed necessary by Operations staff.

Links may be added at any stopping location in a schedule, though will often be added at the destination to provide a continuation of a service that is itself terminating.

Prior to version 11 of the data schema, Links are notified as a pair of new values in the association *CategoryType* enumeration. Two values are provided to allow the direction of a link to be inferred. A link association may be a *link-to* ("LT") or *link-from* ("LF"), indicating the direction of passenger flow from one service to another. If service A has a *link-to* association with service B, then passengers change from service A to service B. Similarly, if service B has a corresponding *link-from* association with service A, then passengers also change from service A to service B.

From version 11 of the data schema, only a single Link association category exists, "LK". A link association is still considered a *link-to* or *link-from*, but this is inferred from the main/associated service identifiers in the *association* message. Passengers always travel from the main service onto the associated service, thus a *link-to* association exists from the main service to the associated service. A *link-from* association exists from the associated service to the main service.

Where a *link-to* association appears at the destination of a service and the corresponding *link-from* association appears at the origin of the linked service, this is an *end-to-end* link, which should be interpreted as a simple combination of the two services into a single "logical" service.

If a link association appears where it is not *end-to-end*, then this should be interpreted in a way similar to a divide or join association, in that it gives the possibility of multiple origins and/or destinations from a given calling point in a schedule. However, note the possibility that the linked service may involve a change to another type of transport, e.g. from a train to a bus, which does not happen with "normal" divide or join associations.

4.3.6 Actual and Forecast Information (the 'TS' element)

"Actual and Forecast Information" messages convey "real time" information related to stations in the train's schedule. These messages are not used to communicate changes in schedules (which will be communicated by Schedule Messages), but will convey information about:

- Expected times of arrival and departure at calling points
- Expected times of passing at passing points (from v9 onwards)
- Unknown delay status indication ("delayed")
- Current manual delay values (from version 11 onwards)
- Actual times of arrival and departure at calling points
- Actual times of passing at passing points (from v9 onwards)
- Information regarding the source of an expected or actual time (from v8 onwards)
- Platform numbers at calling points (including source and confirmation status from version 11 onwards)
- Whether a platform number is *suppressed*, i.e. should not be displayed to the Public
- Whether the service is *suppressed*, i.e. should not be displayed to the Public (from version 11 onwards)
- Train length (from version 11 onwards)
- Whether a train that divides is running in reverse formation at the divide location (from version 11 onwards)
- Whether a train that detaches stock detaches from the front (from version 11 onwards)
- Late running reason

Darwin provides the following information for a train, as it becomes known to Darwin:

- Actual Time – Actual times are provided to Output Ports as a part of the update process, as Darwin becomes aware of them. Note that actual times are not received for all stations. In these instances, the client must consider missing actual times prior to locations with actual times as being "No Report".

  Actual times may be withdrawn, if an operator determines that they have been made in error. In this case, a *TS/Location* element will be sent where the actual time is absent and an estimated time is provided. Additionally, an "atRemoved" attribute will be supplied to indicate that the actual time has been explicitly withdrawn. Note that this "atRemoved" attribute will only be sent once, when the actual time is withdrawn, and is intended only to avoid a race-condition for DCIS clients. It will not be set on subsequent updates or snapshots.

- Forecasts – Darwin supplies expected times of arrival and/or departure as available. To conserve bandwidth, Darwin will only send Actual and Forecast Information for stations where the information has changed. An update message will convey information for a minimum of one station/TIPLOC. Depending on the changes being conveyed, the stations for which information is conveyed need not be contiguous in the train's schedule. Forecast information is provided to Push Ports, where there is a *significant change* to the information that has previously been provided to the Ports, as a part of a snapshot or update.

  Forecast times for locations with Public activities are calculated with reference to the Public Schedule times at those locations. For passing and operational locations, the forecast times are calculated with reference to the Working Schedule times.

  From version 12, where a location has a Public activity and the forecast calculated with reference to the Public Schedule times differs from a forecast calculated with reference to the Working Schedule times, a separate forecast time shall be published giving the value of that "working" forecast. These working forecasts are

intended for operational use only and are not to be displayed in public visible information.

- Forecast or Actual Time Source – From v8 of the data schema onwards, each Forecast or Actual Time will be accompanied by a source attribute (*src*), which indicates from where the associated time originated. Its value is a short string, such as (but not limited to) “Darwin”, “CIS”, “TRUST”, “TD”, “Tyrell”, etc. Note that if in the future Darwin accepts data from other sources, new string values may be returned without a schema update.

  If the source can have multiple instances (currently only for the “CIS” and “TRUST” sources, but this is not a limitation), there may also be supplied a *srcInst* attribute. This attribute is set to a 4 character coded value that indicates which instance of the source set the forecast or actual time. This is currently used to distinguish multiple CIS systems from one another, or the type of TRUST movement that has been reported, i.e. Automatic, Manual or GPS.

  For example, if the *src* attribute is set to “CIS”, there may be a *srcInst* attribute with the value “AM01”. Another forecast may have a *src* attribute set to “CIS”, but a *srcInst* attribute with the value “TH01”. These two forecasts are set by the “Southern Metropolitan” CIS and the “Southeastern” CIS, respectively. A mapping table between the source instance codes and an expanded display name can be downloaded in the Timetable Reference data file from v2 onwards (see sec. 5.3 below).

A *significant change* is defined as:

- A change in the nature of the information that is available for a location, where the nature may be: ‘No Data Available’, no report, a forecast or an actual time.

- The forecast time for the location (arrival, departure or pass) differs to that last reported to the Output Ports by a minute or more.

When the Push Port outputs an Update for a <Location> element, all attributes and child elements defined by the schema will be either explicitly supplied in the message, or can be assumed to take their default value as defined in the schema. If an optional attribute is not supplied and the schema does not specify a default, then it should be assumed to be “unset”.

Prior to version 11 of the data schema, the *TS*/*Location* element provided a “deleted” attribute. Clients should **not** rely on this flag to determine the deletion status of a service. Changes to the deletion status should only be made on receipt of a “Schedule” message. Later schema versions have removed the “deleted” attribute from the *TS*/*Location* element.

4.3.6.1 Use of Platform Numbers

Current platform numbers may be provided in *TS* updates. Although such platform numbers are supplied, they are not always to be used when displaying data to the public.

To determine whether a platform number should be visible to the public, a *platsup* attribute is provided in the *TS*/*Location*/*plat* element (or *TS*/*Location* element prior to

version 11). If this attribute is set to the value "true" then the platform number at that location should not be displayed.

Additionally, two other data items are provided related to platform suppression:

1. If the *cisPlatsup* attribute is set to the value "true" in the *TS*/*Location*/*plat* element then the platform number at that location has been suppressed by a CIS or Darwin Workstation.

2. If the *TS*/*Location*/*suppr* element has the value “true” then the service is suppressed at the location and the platform number at that location should not be displayed.

Either or both of these values being set to “true” shall cause the *platsup* attribute to also be set to “true”. A client that only wants to know whether platforms should be displayed to the public need only check the value of the *platsup* attribute.

4.3.7 Train Order (the ‘trainOrder’ element)

From the version 11 of the data schema, Darwin may publish messages indicating the expected order that trains are to depart from a station platform.

The station is identified by both TIPLOC and CRS code. The TIPLOC will be the TIPLOC supplied by the provider of the train order and may not be the same as the TIPLOC used in the schedules referenced in the message. However, the TIPLOC and those referenced in the schedules will share the same CRS code, as defined in Darwin’s configuration data.

Each message for station platform may set a new train order, or may clear the previous train order. If a new order is set, it completely replaces any previous order.

Up to three trains can be set in the train order. Trains are identified by the Darwin RID identifier, where the train is known by Darwin and an RID has been provided. However, where a Darwin RID identifier has not been provided, or a train is unknown to Darwin, only a Train ID (headcode) will be provided.

When a Darwin RID is given for a train, a set of attributes containing scheduled times is provided to identify the exact schedule location for circular routes.

Where associations occur at a platform where the train order is set, the train order reflects the order that the services depart. Thus, for a service that divides at the platform, the train order separately lists each portion that departs. Where services join, only the single joined service is listed.

4.3.8 Train Table Suppression Status and Station Messages (the ‘OW’ element)

As a part of a **Snapshot (Standard or Full)**, Darwin identifies stations that have their train table suppressed and/or a message assigned, together with the text of the message.

As a part of an **Update**, Darwin provides new and revised details for these items, as information becomes known to Darwin.

A message is attached to a list of stations and includes details on the category (cat) and severity (sev) of the message, plus whether the train running information table normally displayed to the public is to be suppressed for stations to which this message is attached.

The categories and severities that can be applied are detailed here:

Category Code	Meaning
Train	Something that affects the trains calling at the station.
Station	Something related to the station itself such as lifts, escalators, etc.
Connections	Connecting services, for example London Underground.
System	Darwin Systems related.
Misc	Miscellaneous (anything not covered by other categories).
PriorTrains	Advance notices affecting trains, such as engineering work.
PriorOthers	Advance notices affecting other things, such as the lifts being out of order for the next week.

Severity Code	Meaning
0	Situation normal. Message is for reassurance only.
1	A minor item.
2	A major item.
3	A severe item.

Note that the message text can also include two particular basic HTML-like elements. Paragraph elements (“<p>”) are used to break lines, and anchor elements (“<a>”) define links to a URL. Messages can also include character entity references (e.g. “ ”) that some client systems may need to translate before rendering the supplied text.

The message includes an ‘id’ attribute that must be used to match up messages. As the operator is at liberty to change the text of the message, and even have different messages with the same text, this is the only means that should be used to identify messages. The value of ‘id’ is an integer with an opaque meaning. All IDs stored by clients must be cleared when the database is unavailable, a new snapshot is requested or a timetable rebuild is signalled.

Note that the Station elements can be missing, implying that this message has been removed from display at all stations, though the message still exists in the system. Each time a message is received, the full list of stations is supplied to which the message is attached. This implies that removal of a message from a station shall be inferred from the absence of that station in the supplied list.

4.3.9 Overdue Information (the ‘Overdue’ element)

For clients using schemas prior to version 11:

The Push Ports, from the release of version 11 of the data schema, no longer support notification of Overdue trains, the “expired” status and “uncertain” forecast status. Schemas prior to version 11 still have properties for these values, but they will no longer

be populated. The “Overdue” message provided in these schema version will no longer be sent.

For clients using version 11 schema or later :

Overdue trains, the “expired” status and “uncertain” forecast status are no longer provided in the data schema.

4.3.10 Train Alerts (the ‘trainAlert’ element)

The NRCC or a TOC can send important announcements (textual information known as train alerts) out via the Push Port. These alerts can be attached to various services and individual stations called at by those services. Prior to version 11 of the data schema these were known as “Ad-hoc alerts”.

As a part of a **Snapshot (Standard or Full)**, Darwin identifies train alerts and the services and stations to which they apply.

As a part of an **Update**, Darwin provides new and revised versions of these alerts, as information becomes known to Darwin.

A part of a Train alert, the following data items are supplied:

- *Alert ID* – A unique identifier for this alert.
- *Services* – A list of services this alert is attached to.
- *SendAlertBySMS* – (known as *SendAlert* prior to v8) – If set to True then this alert is eligible to send by SMS.
- *SendAlertByEmail* – (v8 onwards) – If set to True then this alert is eligible to send by email.
- *SendAlertByTwitter* – (v8 onwards) – If set to True then this alert is eligible to send by Twitter.
- *Source* – The sender of this alert. The value can be a TOC code, or the string “NRCC”.
- *AlertText* – The text of the alert. See below for details on how to process this field.
- *Audience* – The intended audience for this alert. It may be set to *Customer*, *Staff* or *Operations*.
- *AlertType* – The type of the alert, which may be *Normal* or *Forced*. How a client should differentiate between *Normal* and *Forced* is specific to a client and Darwin does not provide any interpretation. However, a client might typically send *Forced* alerts immediately, whereas *Normal* alerts are only sent within a time window.
- *CopiedFromAlertID* – (optional) – The original *Alert ID* from which this alert has been copied.
- *CopiedFromSource* – (optional) – The *Source* (TOC code or “NRCC”) of the alert from which this alert has been copied.

Prior to v8, the AlertText item was defined to be a simple text string containing the alert message. From v8 onwards, this field has been re-defined to contain an XML-encoded XHTML fragment that may contain embedded paragraph (<p>) and anchor (<a>) tags, as well as non-break space entities ( ). This allows alert messages to contain paragraphs and hyperlinks.

Thus, for example, the following alert message:

> There are no services currently running between London Euston and Manchester Piccadilly.
>
> For further details, please access the National Rail Enquiries website.

will be encoded as:

```
<AlertText>&lt;p&gt;There are no services currently running between London Euston and
Manchester Piccadilly.&lt;/p&gt;&lt;p&gt;For further details, please access the &lt;a
href="http://www.nationalrail.co.uk"&gt;National Rail Enquiries
website&lt;/a&gt;.&lt;/p&gt;</AlertText>
```

Clients whose output mechanism is HTML may directly output the alert text, but others will need to decode the message and convert the paragraphs and hyperlinks into a format suitable for their output medium.

4.3.11 Tracking ID Corrections (the 'trackingID' element)

From version 11 of the data schema, Darwin may publish messages to provide a corrected *tracking ID* (headcode) for a mis-identified train that is being reported by TD.NET.

The "trackingID" message provides the TD berth where the train is currently expected to be, the original incorrect headcode, and the new correct headcode that should be used to track the service.

Tracking ID corrections are only sent to the originator of the request and those clients that have requested *TD filtering* (see sec. 4.2.3) of the TD area that contains the incorrectly tracked service.

These messages will only be published once, when the corrected tracking ID becomes known to Darwin. They will not be included in any snapshot.

4.3.12 Alarms (the 'alarm' element)

From version 11 of the data schema, Darwin may publish messages related to various alarm conditions.

The conditions that shall generate an alarm are:

Alarm	Description
TD Area failure	Darwin has detected that no data has been received from a single TD area for a defined period of time.
TD Feed failure	Darwin has detected that no data has been received

	from any TD area for a defined period of time.
Tyrell Feed failure	Darwin has detected that its connection to Tyrell has failed.

Not all clients will be configured to receive all, or any, alarms. Additionally, only those clients that have requested *TD filtering* (see sec. 4.2.3) of the applicable TD area will receive TD Area failure-related alarms.

Each alarm has a unique identifier associated with it. An alarm may be cleared by an "alarm" message containing a "clear" element that has the unique identifier of the alarm to be cleared.

Snapshots contain only those alarms that are currently active.

5. Timetable Files

Darwin makes available timetable files to registered clients via FTP. A Timetable ID is used to identify the currently available timetable files. By using the Timetable ID Query (see Section 6.1.2.2) to determine the timetable ID and file names, clients are able to determine whether the timetable has changed since their last enquiry. The client is then able to download new files using FTP. The following files are available to registered clients:

- **Timetable Data**. The set of ITPS-derived schedules covering at least a 48-hour period held in the Darwin database. This list of schedules provides the basis on which a Darwin snapshot can be applied. The schedules in the timetable do not include forecast or actual times although they reflect the latest state that Darwin has when the timetable file was generated, so any schedule changes, new schedules, false destinations, cancellations and associations will be included. See Section 5.2 for details of the file format. Note that the schedules also include platform numbers. However, this platform number is the one that was current when the timetable file was generated. This will typically be the scheduled platform. The *current* platform number may be different and if so will be transmitted in a snapshot update.

- **Timetable Reference Data**. Reference data in the form of mappings from the TIPLOC to CRS, TOC and Name, reason codes and via texts. See Section 5.3 for details of the file format.

Note: A client must not retrieve timetable data without first enquiring the current timetable ID.

The FTP of timetable data is expected to take significant time (see Section 8 for estimates). Darwin is able to send regular updates whilst a Client is downloading the timetable but the updates sent will be in relation to the new timetable and may not be compatible with the previous timetable.

Once a Client has downloaded a new timetable, the Client should clear out the old timetable, initialise the new timetable and then request a snapshot. This ensures that it is fully up to date with Darwin.

5.1 Timetable Files available by FTP

A client registered as requiring timetable data through the Push Ports interface is able to retrieve the current timetable from the Darwin System.

Once connected the client has read access to two directories:

`/timetable`	This directory holds the XML timetable files.
`/snapshot`	This directory holds snapshot data if it was requested via FTP.

Darwin provides access to its current timetable via FTP. Clients use the Timetable ID query to get the ID of the current timetable so the client can identify if the timetable matches the version it currently holds. The Timetable ID query also returns the filenames of the current timetable and reference data files for the schema version the client is using.

The client can download the current timetable and reference data files from the Darwin servers using FTP.

Timetable files are available for transfer from Darwin providing:

1. A list of ITPS derived schedules.

2. A table to translate station representations (held as CRS codes and TIPLOCs) to plain English.

3. A table to translate TOC codes to plain English.

4. A table to translate late running reason codes and cancellation reason codes to plain English.

5. A table to allow the client to append “via” text to the destination of a train.

6. A table to translate Darwin CIS Instance codes to display strings.

Further details of these timetable data files are given below.

Details of the FTP account provided are in section 3.2.

5.2 Timetable Data

The timetable data file will be compressed using gzip. The timetable file name is returned in the response to the timetable ID query and is dependent on the timetable schema version requested by the client when it initially connects. The example below shows a single journey from Manchester Airport to Sheffield with an associated journey (not shown) at Manchester Piccadilly.

```
<?xml version="1.0" encoding="utf-8"?>
<PportTimetable xmlns="http://www.thalesgroup.com/rtti/XmlTimetable/v8"
           timetableID="20140619030427">
   <Journey rid="201406190276527" uid="C75596" ssd="2014-06-19" trainId="1B72" toc="TP">
      <OR tpl="MNCRIAP" act="TB" wtd="10:00:00" ptd="09:59"/>
      <IP tpl="MNCRPIC" act="T " wta="10:13:00" wtd="10:19:00" pta="10:13" ptd="10:19"/>
      <PP tpl="ARDWCKJ" wtp="10:21:00"/>
      <IP tpl="STKP" act="T " wta="10:29:00" wtd="10:29:30" pta="10:29" ptd="10:29"/>
      <DT tpl="SHEFFLD" act="TF" wta="11:07:00" pta="11:08"/>
   </Journey>
   <Association tpl="MNCRPIC" cat="VV">
      <main rid="20140619030427" wta="10:13:00" wtd="10:19:00" pta="10:13" ptd="10:19"/>
      <assoc rid="201406190304499" wtd="10:25:00" ptd="10:25"/>
   </Association>
</PportTimetable>
```

Multiple versions of the timetable data schema are supported from Version 4. Version 5 is an update that includes platform data and details of cancelled locations. Version 6 adds support for Linked Associations. Version 7 adds support for passing points and working times. Version 8 adds support for DCIS data, as introduced by data schema v11. The latest schema versions can be found in section 7.1. Clients may use any supported version, according to their needs, but are recommended to upgrade to the latest version when the opportunity presents itself.

Note that also from version 7, the definition of schedule activity codes has changed slightly. Previously, they consisted of 12 character fixed length data, reflecting the way they are provided from CIF. From version 7 onwards, activities have been re-defined to be

6x2 character data, with empty activities consisting of two spaces (" ") removed. Thus a typical activity of "T " has changed to "T ".

5.2.1 Q Trains

The Push Port XML timetable supports services that run "as required" (Q Trains). Q Trains are journeys that are in the timetable, but are not scheduled to run. Darwin treats these trains as non-existent until they are activated, either by an explicit activation request, or a movement or forecast report. Once a Q Train becomes active, the schedule is sent out to Push Port clients and updates happen from then on as for any other journey. Darwin will not automatically activate a Q Train until an external update is received for it.

These journeys are also provided to subscribers to the Push Ports XML timetable. So that clients can differentiate these particular journeys, a boolean attribute is set true (qtrain) to indicate that this journey is not scheduled to run.

For example:

```
<?xml version="1.0" encoding="utf-8"?>
<PportTimetable xmlns="http://www.thalesgroup.com/rtti/XmlTimetable/v8"
            timetableID="20140619030427">
   <Journey rid="201406190276127" uid="Q00001" ssd="2014-06-19" trainId="1Q11" toc="NW"
            qtrain="true">
      <OR tpl="MNCRPIC" wtd="10:00:00" ptd="09:59" act="TB"/>
      <DT tpl="OLDHAMW" wta="10:13:00" pta="10:15" act="TF"/>
   </Journey>
</PportTimetable>
```

5.3 Timetable Reference Data

The timetable reference data file is compressed using gzip, and the file name is returned as part of the timetable ID query response. The TIPLOC reference data provides additional information for the locations listed in the timetable data file, including information on the mapping between TOC codes and the full English name.

Also provided is a list of the reason codes and their associated text, for both late running reasons and cancellations.

Next is a list of via texts and their criteria for invocation. Note that the sequence of "Via" statements is significant, with the most significant entries supplied first. A versioned namespace is used to prevent clients getting confused by incompatible schema versions.

From v2 of the Timetable Reference Data schema, a list of CIS Source code to name mappings is provided. These map the forecast source instance code values used in forecast updates to displayable strings. Clients that do not wish to use this data may continue to use v1 of the schema, though are recommended to update to use the latest version when the opportunity presents itself.

5.3.1 Example Reference Data

```
<?xml version="1.0" encoding="utf-8"?>
<PportTimetableRef
      xmlns="http://www.thalesgroup.com/rtti/XmlTimetable/v2/rttiCTTReferenceSchema.xsd"
      timetableId="20050121105940">
   <LocationRef tpl="MNCROXR" crs="MCO" toc="NW" name="Manchester Oxford Road"/>
   <LocationRef tpl="MNCRPIC" crs="MAN" toc="RT" name="Manchester Piccadilly"/>
```

```
   <LocationRef tpl="MOSSLEY" crs="MSL" toc="NW" name="Mossley"/>
   <LocationRef tpl="OLDHAMM" crs="OLM" toc="NW" name="Oldham Mumps"/>
   <TocRef toc="NW" name="First North Western" url="http://www.url.com" />
   <TocRef toc="RT" name="Network Rail" url="http://www.url.com" />
   <LateRunningReasons>
      <Reason code="1" reasontext="This train is delayed due to a problem"/>
   </LateRunningReasons>
   <CancellationReasons>
      <Reason code="1" reasontext="This train has been cancelled due to a problem"/>
   </CancellationReasons>
   <CISSource code="TH01" name="Southeastern"/>
   <CISSource code="AM01" name="Southern Metropolitan"/>
   <CISSource code="LI01" name="York "/>
</PportTimetableRef>
```

6. Example Client Requests and Darwin Responses

6.1 Client Requests

Darwin supports the following Setup Phase requests from clients see section 6.1.1.

- Request Schema Versions
- Connect

Darwin supports the following Data Phase requests from clients see section 6.1.2.

- Timetable ID Query
- Snapshot Requests
- Update Requests

6.1.1 Setup Phase Requests

Note that the PPSetupReq element is defined to allow *any* child elements. This allows for future extension of the protocol but does mean that XML validation is meaningless, as any well-formed XML child elements will be valid.

6.1.1.1 Request Versions

This is the first message a client will send after connecting to the push port server socket and indicates to the server what versions of the data, timetable and reference data schemas the client will use.

```
<?xml version="1.0" encoding="utf-8"?>
<PPSetupReq xmlns="http://thalesgroup.com/RTTI/PushPortSetup/root_1">
  <PPReqVersion xmlns="http://thalesgroup.com/RTTI/PushPortStatus/root_1"
        version="http://www.thalesgroup.com/rtti/PushPort/v9"
        ttversion="http://www.thalesgroup.com/rtti/XmlTimetable/v7/rttiCTTSchema.xsd"
        ttrefversion="http://www.thales-
is.com/rtti/XmlTimetable/v2/rttiCTTReferenceSchema.xsd" />
</PPSetupReq>
```

Note that the versions given in the example above are not necessarily the most current or most suitable for any particular client. A client must obtain the correct values from the exact version of the schema files that they wish to use.

Any attempt to send any other message before a request versions message will result in an INVREQ status message response.

Any attempt to request versions that are not supported by the push port server will result in an INVVER status message response.

6.1.1.2 Request Filtering

This is an optional message that a client may send to request a *filtered* push port. The message identifies the station locations for which filtered data is required.

```
<?xml version="1.0" encoding="UTF-8"?>
<PPSetupReq xmlns="http://thalesgroup.com/RTTI/PushPortSetup/root_1"
xmlns:pp1="http://thalesgroup.com/RTTI/PushPortFilter/root_1">
```

```
   <pp1:FilterTiplocs>
      <pp1:tiploc>MNCRPIC</pp1:tiploc>
      <pp1:tiploc>STKP</pp1:tiploc>
      <pp1:tiploc>GATLEY</pp1:tiploc>
   </pp1:FilterTiplocs>
</PPSetupReq>
```

If filtering is requested, it must be acknowledged with a HBOK status message in order for a subsequent Connect request to succeed.

6.1.1.3 Connect

This indicates to the Push Port server that the setup phase is complete for this client and that this client will now start using messages defined within the requested data schema version, these request are described in section 6.1.2.

```
<?xml version="1.0" encoding="utf-8"?>
<PPSetupReq xmlns="http://thalesgroup.com/RTTI/PushPortSetup/root_1">
  <PPConnect xmlns="http://thalesgroup.com/RTTI/PushPortStatus/root_1" />
</PPSetupReq>
```

Any attempt to connect before a valid request versions message will result in an INVREQ status message response.

6.1.2 Data Phase Requests

6.1.2.1 Push Port Data Requests Common Part

All Darwin data requests are wrapped in a **<Pport>** element. This element includes a timestamp, the version of the data schema used and the required namespace directives. An example is provided here:

```
<?xml version="1.0" encoding="UTF-8"?>
<Pport ts="2010-05-21T11:00:37" version="11.0" xmlns="http://www.thalesgroup.com/rtti/PushPort/v11">
```

The version attribute should be set to the version number of the data schema used (see section 7.1) and the ts (timestamp) should be set to your current local time.

In the following examples, the common attributes in **<Pport>** elements are omitted for clarity.

6.1.2.2 Timetable ID Query

This allows the Client to query for the current timetable identifier and the names of the timetable and reference data files that are available for download. The filenames returned are dependent on the timetable and reference data versions requested by the client when it connects.

The Client can request the timetable identifier at anytime.

```
<?xml version="1.0"?>
<<Pport …>
   <QueryTimetable />
</Pport>
```

If a timetable rebuild is ongoing or Darwin is unavailable then Darwin will respond with a HBFAIL status message.

6.1.2.3 Snapshot Request

Allows the client system to request a snapshot of train information, as described in section 4.3.3. The standard snapshot request is:

```
<?xml version="1.0"?>
<Pport …>
   <GetSnapshotReq />
</Pport>
```

The full snapshot request is:

```
<?xml version="1.0"?>
<Pport …>
   <GetFullSnapshotReq />
</Pport>
```

Note that either of the snapshots can request that data be made available via FTP as an XML file compressed using gzip:

```
<?xml version="1.0"?>
<Pport …>
   <GetSnapshotReq viaftp="true"/>
</Pport>

<?xml version="1.0"?>
<Pport …>
   <GetFullSnapshotReq viaftp="true"/>
</Pport>
```

6.1.2.4 Start Update Request

This message will trigger the start of updates from Darwin. Updates include any updates to any of the journeys held in Darwin including movement reports, schedule changes, etc. See section 4.3.1 for an overview of the data sent.

```
<?xml version="1.0"?>
<Pport …>
   <StartUpdateReq/>
</Pport>
```

6.1.2.5 Stop Update Request

This message is used to stop the updates from Darwin.

```
<?xml version="1.0"?>
<Pport …>
   <StopUpdateReq/>
</Pport>
```

6.2 Darwin Responses

Due to the volume of data to be transmitted, the data provided by Darwin is structured to minimise bandwidth. This has led to the use of abbreviations in the choice of tag and attribute names.

All messages are time-stamped on transmission with the current time (i.e. the time taken from Darwin's internal clock). The messages conform to the W3C XML definitions.

There is no indication of GMT/BST in the data provided. This is intentional as it is not possible to reliably establish whether the data flowing into Darwin is GMT or BST. Times should be assumed to be local time.

The following sections detail the responses from Darwin to the requests made by the Client as described in Section 6.1).

6.2.1 Darwin Data Response Common Parts

All Darwin data responses are wrapped in a **<Pport>** element. This element includes a timestamp, the version of the schema used on the server and the required namespace directive. An example is provided here:

```
<?xml version="1.0" encoding="UTF-8"?>
<Pport ts="2010-05-21T11:00:37" version="11.0" xmlns="http://www.thalesgroup.com/rtti/PushPort/v11">
```

The version attribute will be set to the version number in the schema definition (see section 7.1) and the ts (timestamp) will be set to the local time on the Darwin system.

In the following examples the common attributes in **<Pport>** elements are omitted for clarity.

6.2.2 Timetable Identifier

The timetable identifier and timetable/reference data filenames will be communicated via a Timetable ID element.

```
<TimeTableId ttfile="20100524033206_v7.xml.gz"
             ttreffile="20100524033206_ref_v1.xml.gz">20100524033206</TimeTableId>
```

The `ttfile` and `ttreffile` attributes identify the names of the timetable files that can be downloaded via FTP.

6.2.3 Snapshots

Upon request, Darwin will provide a snapshot to a Client. Further snapshot requests from the same Client shall be rejected until the snapshot completes.

Snapshots will provide the following data:

- Schedule Information ('schedule')
- Actual and Forecast Information ('TS')
- Association information ("association")
- Table Suppression and Station Messages ('OW')
- Train order information ("trainOrder")
- Train Alert Messages ("trainAlert")
- Alarms ('alarm')

Snapshots use the same XML format as updates (see below). Snapshots are presented in a collated form using the <sR> element in place of the <uR> element.

Snapshots can be delivered in two forms, either back over the connection or from the Push Ports FTP server.

6.2.3.1 Snapshots over the Push Ports Connection

In this instance, the resulting XML is sent back over the TCP/IP connection to the client.

6.2.3.2 Snapshots via FTP

If the client requests that the snapshot be retrievable via FTP then the snapshot that would have been sent back is saved to a file and compressed using gzip. A message then is sent indicating the filename that a client should use to fetch the snapshot data.

```
<SnapshotId>filename<SnapshotId/>
```

No assumptions should be made on the format of the filename, though it is guaranteed to contain only alphanumeric characters and periods, and to be unique between clients connected in parallel.

Details of the FTP account provided are in section 3.2.

Since FTP snapshots are compressed, large snapshots may be received significantly quicker than when received over the Push Port socket connection.

Snapshot files are automatically deleted by Darwin, usually daily. A client does not need to (and cannot) delete them itself.

6.2.4 Updates

Updates will provide the following data:

- Schedule Information ('schedule')
- Deactivation ('deactivated')
- Actual and Forecast Information ('TS')
- Association information ('association')
- Table Suppression and Station Messages ('OW')
- Train order information ('trainOrder')
- Train Alert Messages ('trainAlert')
- Tracking ID changes ('trackingID')
- Alarms ('alarm')

6.2.4.1 Schedule Information ('Schedule')

An example of a schedule is as follows:

```
<Pport  xmlns:sch="http://www.thalesgroup.com/rtti/PushPort/Schedules/v1">
    <uR requestSource="CIS1" requestID="X12345678">
        <schedule rid="20030626001234" uid="C09014" trainId="9X99" ssd="2003-06-26" toc="NW" status="P"
trainCat="OO" >
            <sch:OR tpl="MNCRPIC" act="TB" ptd="12:00" wtd="12:01" fd="CREWE  "/>
            <sch:PP tpl="ARDWCKJ" act=" " wtp="12:02"/>
            <sch:IP tpl="CREWE " act="T " pta="12:30" ptd="12:31" wta="12:30" wtd="12:31:30"/>
            <sch:IP tpl="SBCH   " act="T " pta="12:40" ptd="12:45" wta="12:40" wtd="12:45" can="true"/>
            <sch:DT tpl="CHSTR  " act="TF" pta="13:00" wta="13:00"/>
        </schedule>
    </uR>
</Pport>
```

6.2.4.2 Actual and Forecast Information ('TS')

In practice, updates will generally consist of a single location with an actual time recorded followed by locations with forecast times for the rest of the journey. Earlier locations in the journey will only be sent if there is a change to report, such as Darwin receiving an actual time for a location some time after the departure actually occurred.

```
<Pport  xmlns:for="http://www.thalesgroup.com/rtti/PushPort/Forecasts/v1">
    <uR requestSource="CIS1" requestID="X12345678">
    <TS rid="20030626001234">
            <for:LateReason>123</for:LateReason>
            <for:Location tpl="MNCRPIC" wtd="12:01" ptd="12:00">
                <for:dep at="12:05" src="TD"/>
                <for:plat platsrc="A" conf="true">11</for:plat>
            </for:Location>
            <for:Location tpl="ARDWCKJ" wtp="12:02">
                <for:pass et="12:07" src="Darwin"/>
            </for:Location>
            <for:Location tpl="CREWE " wta="12:30" wtd="12:31:30" pta="12:30" ptd="12:31">
                <for:arr et="12:33" src="Darwin"/>
                <for:dep et="12:45" etmin="12:45" src="CIS" srcInst="CIS1"/>
                <for:plat platsup="true" cisPlatsup="true" platsrc="M">4</for:plat>
            </for:Location>
            <for:Location tpl="CHSTR  " wta="13:00" pta="13:00">
                <for:arr et="13:10" src="Darwin"/>
                <for:plat platsrc="P">2</for:plat>
            </for:Location>
        </TS>
    </uR>
</Pport>
```

6.2.4.3 Station Messages ('OW')

An example of a station message is provided below. As noted above, the displayed text can also include paragraph elements ("<p>") and anchor elements ("<a>"), so the content can be quite complex as seen in the following examples:

```
<uR>
   <OW id="123" cat="Misc" sev="1">
      <Station crs="MAN" />
      <Station crs="EUS" />
      <Msg>This is a single line with a <a href="http://host.domain">link</a>.</Msg>
   </OW>
</uR>

<uR>
   <OW id="123" cat="Misc" sev="1" suppress="true">
      <Station crs="MAN" />
      <Msg><p>Line one</p><p>Line two</p></Msg>
   </OW>
</uR>
```

```
<uR>
   <OW id="123" cat="Misc" sev="1" >
      <Station crs="MAN" />
      <Msg><p>Line one</p>
      <p>Line two and a<a href="http://host.domain">link</a></p>
      </Msg>
   </OW>
</uR>

<uR>
   <OW id="123" cat="Misc" sev="1" >
      <Msg>This message is not on display.</Msg>
   </OW>
</uR>
```

6.2.4.4 Train Alert Messages ('trainAlert')

The NRCC or a TOC can send important announcements (textual information known as train alerts) out via the Push Port. These alerts can be attached to various services and individual stations within those services. The following is an example of an train alert.

```
<uR>
   <AdhocAlert>
      <AlertID>1</AlertID>
       <Services>
          <Service UID="P60844" SSD="2010-05-25">
              <Location>EUS</Location>
              <Location>SOT</Location>
              <Location>MAC</Location>
              <Location>SPT</Location>
              <Location>MAN</Location>
          </Service>
       </Services>
      <SendAlertBySMS>true</SendAlertBySMS>
      <SendAlertByEmail>false</SendAlertByEmail>
      <SendAlertByTwitter>false</SendAlertByTwitter>
      <Source>NRCC</Source>
      <AlertText>Adhoc Test Alert 1</AlertText>
      <Audience>Customer</Audience>
      <AlertType>Normal</AlertType>
   </AdhocAlert>
</uR>
```

The Source of a train alert may be a two-character TOC code, or the string "NRCC".

7. XML Schemas

7.1 XSD Files

Due to the number of files that now constitute the whole Push Port schema, and problems that have been encountered when trying to embed a ZIP archive, the XSD files are no longer embedded within this document. The required schema files are included in a separate ZIP archive that also includes this document.

The latest schema versions are:

Schema	Version	Namespace	Filename
Setup	3	http://thalesgroup.com/RTTI/PushPortSetup/root_1	rttiPPTSetup_v3.xsd
Data	12	http://www.thalesgroup.com/rtti/PushPort/v12	rttiPPTSchema_v12.xsd
Timetable	8	http://www.thalesgroup.com/rtti/XmlTimetable/v8	rttiCTTSchema_v8.xsd
Reference Data	3	http://www.thalesgroup.com/rtti/XmlRefData/v3	rttiCTTReferenceSchema_v3.xsd

7.2 Schema Versioning

All the schemas belong to namespaces that include version numbers. The Push Port Service itself supports multiple versions from version 1 of the Setup Schema and version 7 of the Updates Schema onwards and the namespaces are used to indicate the version required. The namespaces are also used as a defence against clients using versions of the schema incompatible with the software.

8. Data Volume

The volume of real time data sent by Darwin is dependent on the number of and the size of the Darwin data sources. This is likely to increase as more data sources are added to Darwin.

Current file sizes are of the order of:

Timetable Data: 16 Mbytes
Timetable Reference Data: 190Kbytes

Assuming a 256kbs link (using 20Kb/sec), the transfer of Timetable Data should take around 14 minutes and the Timetable reference data around 10 seconds.

Snapshots sizes are dependent on when the snapshot is requested. Immediately after a Darwin timetable rebuild snapshots, will be small (in the order of 300Kbytes i.e. 10 seconds). At the end of the day a full snapshot is likely to be of the order of 300 Mbytes (i.e. 2.8 hours).

If using gzip compressed FTP snapshots then a saving of up to 90% on the above figures is possible.

Note that ports filtered on a small number of TIPLOC codes will considerably reduce the data volume of both snapshots and real-time updates. However, timetables are not filtered and will always be a similar size.

9. Abbreviations and Glossary

ATOC	Association of Train Operating Companies
CIF	Common Interface File. The format of this file defines the format in which ITPS provides schedule information.
CRS	Computerised Reservation System
DCIS	Darwin CIS. Interface between Darwin and CIS systems driving passenger displays at stations.
False Destination	A train destination. Typically used in a circular route to provide a route for the train.
FTP	File Transfer Protocol.
gzip	A compression tool using the DEFLATE format as defined in RFC 1951. gzip is defined in RFC 1952.
ITPS	Integrated Train Planning System (replacement source of schedule data for TSDB).
LDB	Live Departure Boards. This is the publicly available web interface to the Darwin system.
RID	Darwin generated ID. A unique ID held within the Darwin database to identify a journey.
RTTI	Real Time Train Information – database of train running information. Previous name for Darwin.
Snapshot	There are two types of snapshot: 1. Standard Snapshot: information for all train journeys in the Darwin database that are in progress or have yet to commence. 2. Full Snapshots: includes information for all journeys in the Darwin database since the last timetable rebuild. I.e. includes the standard snapshot plus historic schedule information. Snapshot data is defined in Section 4.3.1.
Theseus	Theseus supplies train-running data received from TOC's to Darwin.
TIPLOC	Timing Point Location
TOC	Train Operating Company
TSDB	Train Services Database (now superseded by ITPS).
Updates	Darwin provides update information to the Client when this information becomes known to Darwin. Update data is defined in Section 4.3.1.
UID	Unique Identifier. (However, the UID is not always unique within the Darwin database)