

AIMS Web Services (SOAP/WSDL) – Usage

AIMS Web Services is a new feature designed to allow your IT Team to extract and/or update AIMS data directly from your application or a third party application. AIMS Web Services have been built using the Simple Object Access Protocol (SOAP) 1.1. and it's currently a library of approximately a dozen different flexible services (for more information about the SOAP Protocol please access <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>).

The AIMS Web Services also provide a Web Services Description Language (WSDL) XML interface which describes the Web Services provided by AIMS and how to access the data.

The following topics are covered in this order:

- **How to access the AIMS Web Service**
- **List of Web Services currently provided by AIMS**

How to access the AIMS Web Service

Notes

- You will need to have a Programming Tool that supports the SOAP/WSDL XML protocols like Microsoft Visual Studio .NET, Delphi, Java or any other tool that supports data Serialization. (for more information please refer to your developing tool help and notes)

How to access the AIMS Web Services WSDL Interface

In order to access the AIMS Web Services Interface you need to have access to the AIMS Web Server contents.

- Locate the following HTTP Address

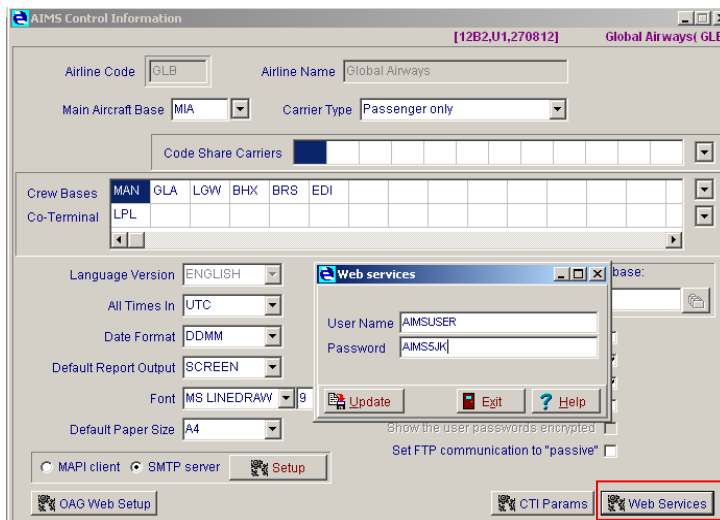
`http://<FQDN>/wtouch/AIMSWebService.exe/wsdl/IAIMSWebService`

Where FQDN = Full Qualified Domain Name

Once the WSDL XML file is imported you will find all the functions of the above list as well as the supported Data Types. Your developing tool should be able to create the client's necessary data structures and functions that will allow you to collect the data. You should find all the public functions listed above under the AIMSWebService Interface.

Authentication

In order to access data and use this functionality, you must first setup the Authentication details through AIMS Option 7.1 (Web Services button).



The credentials above are required whenever a function that is being exposed by Web Services is being used / called.

Those credentials are being saved in the database being used by AIMS. That means that you may have set them up differently per each database copy.

Without credentials being set Web Services functions can't be used.

List of Web Services currently provided by AIMS

Click on each link below to take you to the associated Method (function). Field list information is also provided towards the end of the document (also linked to each function).

Methods.....	4
Crew Related.....	4
1. List of Crew Members operating a flight leg.....	4
2. List of Crew Members operating flights of a specific day.....	4
3. List of Crew Members operating flights of a specific day and time period.	5
4. List of Crew Members operating a specific Crew Pairing on a specific day.....	5
5. Crew Schedule Changes in a given period	5
6. List of Unfinalized (Disrupted) Crew Pairings for a given day.....	6
7. List of crew with missing training indicators for a given period.....	6
8. Insert new crew member in AIMS Crew Database (Ver. 1).....	6
9. Insert new crew member in AIMS Crew Database (Ver. 2).....	6
10. List of pending notifications for a crew for a given period	7
11. Remove Crew Notification Indicator on a certain date	7
12. List crew personal information (inc Quals) for a specific crew or a group for a given period	8
13. List Personal Crew Schedule information for a specific crew for a given period	8
14. List Crew Qualifications for a specific crew or a group for a given period	9
15. Record fact that crew has checked in and provide details of the duty	10
16. Records fact that crew has checked out and provides details of the duty	10
17. List of Crew Members and their Max FDP values on a flight leg.....	11
18. List of Crew Messages.....	11
Flight Related.....	12
19. Flight Details for all flights of a specific day	12
20. Flight Details for all flights in the selected date/time range	12
21. Flight details for specific flights for the selected date/time range	13
22. Flight Schedule changes for a given period.....	13
23. Flight Schedule Modification Log for a given period	14
24. Upload Journey Log data for a specific flight leg.....	15
25. List of Aircraft Routes with its flight legs for a given period.....	15
26. Upload various leg information for a specific flight leg.....	16
Miscellaneous Web Services	16
27. List of Aircraft	16
28. List of Aircraft Types.....	16
29. List of Countries	17
30. List of Airports.....	17
31. List of Charterers	18
32. Time difference between UTC/Local for a station for a given date and time	18
Field Lists.....	19

Methods

Crew Related

1. List of Crew Members operating a flight leg

The function **FetchLegMembers** returns the crew members found on a specific 'Flight' that has been operated on a specific day.

function FetchLegMembers(const UN : WideString; const PSW : WideString; const DD : WideString; const MM : WideString; const YY : WideString; const Flight : WideString; const Dep : WideString): TAIMSGetLegMembers;		
TAIMSGetLegMembers = class(TRemotable)		
FlightDD	WideString	
FlightMM	WideString	
FlightYY	WideString	
FlightCarrier	WideString	
FlightNo	Integer	
FlightLegCD	WideString	
FlightDesc	WideString	
FlightDep	WideString	
Fcount	Integer	Members found on that day
Ferrorexplanation	WideString	Message if an error occurs
Fmember	Tmemberlist	Tmemberlist = array of TAIMSMember ;

2. List of Crew Members operating flights of a specific day

The function **FetchLegMembersPerDay** returns all the Flights and their corresponding crew members which has been operated on a specific day.

function FetchLegMembersPerDay(const UN : WideString; const PSW : WideString; const DD : WideString; const MM : WideString; const YY : WideString): TAIMSDayGetLegMembers;		
TAIMSDayGetLegMembers = class(TRemotable)		
DD	WideString	
MM	WideString	
YY	WideString	
Day	Smallint	
count	Integer	
Ferrorexplanation	WideString	Message if an error occurs
daymember	Tdaymemberlist;	= array of TAIMSGetLegMembers

3. List of Crew Members operating flights of a specific day and time period.

The function **FetchLegMembersForPeriod** returns all the Flights (and their corresponding crew members) which have been operated on a specific day and have departed in the requested time range of that day.

function FetchLegMembersForPeriod(const UN : WideString; const PSW : WideString; const DD : WideString; const MM : WideString; const YY : WideString; const FromHHrs : WideString; const FromMMin : WideString; const ToHHrs : WideString; const ToMMin : WideString): TAIMSPeriodGetLegMembers;		
TAIMSPeriodGetLegMembers = class(TRemotable)		
DD	WideString	Requested Day (DD)
MM	WideString	Requested Month (MM)
YY	WideString	Requested Year (YYYY)
Day	Smallint	AIMS value of the requested date
FromHrs	WideString	Starting hour(s) on the requested day
FromMin	WideString	Starting minute(s) on the requested day
ToHrs	WideString	Ending hour(s) on the requested day
ToMin	WideString	Ending minute(s) on the requested day
Count	Integer	How many Flights found on the date/time
ErrorExplanation	WideString	Message if an error occurs
DayMember	TDayMemberList;	= array of TAIMSGetLegMembers

4. List of Crew Members operating a specific Crew Pairing on a specific day

The function **FetchCrMembers** returns the crew members for a Crew Route on a specific day.

function FetchCrMembers(const UN : WideString; const PSW : WideString; const DD : WideString; const MM : WideString; const YY : WideString; const Croute : WideString): TAIMSCrMembers;		
TAIMSCrMembers = class(TRemotable)		
CrouteDD	WideString	
CrouteMM	WideString	
CrouteYY	WideString	
count	Integer	
Ferrorexplanation	WideString	Message if an error occurs
Fmember	Tmemberlist	= array of TAIMSMember ;

5. Crew Schedule Changes in a given period

The function **CrewScheduleChangesForPeriod** returns the Log records found for a specific date range.

function CrewScheduleChangesForPeriod(const UN : WideString; const PSW : WideString; const FromDD : WideString; const FromMM : WideString; const FromYYYY : WideString; const ToDD : WideString; const TOMM : WideString; const TOYYYY : WideString): TAIMSCrewLogList;		
TAIMSCrewLogList = class(TRemotable)		
CrewLogList	TCrewLogList	= array of TAIMSCrewLogltn ;
ErrorExplanation	WideString	
LogCounter	Integer	

Refer to Field List 7

6. List of Unfinalized (Disrupted) Crew Pairings for a given day

The function **UnfinalizedCrewRoutesForDay** returns a list containing the Crew Routes that are unfinalized for a specific day.

function UnfinalizedCrewRoutesForDay(const UN : WideString; const PSW : WideString; const DD : WideString; const MM : WideString; const YY : WideString): TAIMSUnfinCrewRoutesList;		
TAIMSUnfinCrewRoutesList = class(TRemotable)		
UnfinRteCount	Integer	
UnfinRteList	TUnfinRteList	= array of TAIMSUnfinRteItem ;
ErrorExplanation	WideString	

7. List of crew with missing training indicators for a given period

The function **CrewMissingTrnIndicatorsForPeriod** returns a list of crew with missing training indicators for a given period.

function CrewMissingTrnIndicatorsForPeriod(const UN : WideString; const PSW : WideString; const FmDD : WideString; const FmMM : WideString; const FmYY : WideString; const ToDD : WideString; const ToMM : WideString; const ToYY : WideString): TAIMSCrwMissTrnIndctrList;		
TAIMSCrwMissTrnIndctrList = class(TRemotable)		
CrwMissTrnIndCount	Integer	
CrwMissTrnIndList	TCrwMissTrnIndList	= array of TAIMSCrwMissTrnIndItem ;
ErrorExplanation	WideString	

8. Insert new crew member in AIMS Crew Database (Ver. 1)

The function **InsertCrewMember** attempts to insert a new crew member to the crew database file with details that applied to the TAIMSCrewInfo parameter object e.g. ID, Name, Qualification details. On success it will return a True value, otherwise it returns False and the "Error" explanation.

function InsertCrewMember(const UN : WideString; const PSW : WideString; const CrewInfo : TAIMSCrewInfo; var Error : WideString): Boolean;	
---	--

9. Insert new crew member in AIMS Crew Database (Ver. 2)

The function **InsertCrewMember2** attempts to insert a new crew member to the crew database file (in the same manner as web-service #8 **Insert new crew member in AIMS**) with multiple qualification details and crew categories but with a minimum detail set that is applied to the TAIMSCrewInfo2 parameter object e.g. ID, Name, Gender, Leave Category etc. On success it will return a True value, otherwise it returns False and the "Error" explanation.

function InsertCrewMember2(const UN : WideString; const PSW : WideString; const CrewInfo : TAIMSCrewInfo2; var Error : WideString): Boolean;	
---	--

10. List of pending notifications for a crew for a given period

The function **CrewPendingNotifForPeriod** returns a list of pending notifications for a crew for a given period.

function CrewPendingNotifForPeriod(const UN : WideString; const PSW : WideString; const ID : Integer; const FmDD : WideString; const FmMM : WideString; const FmYY : WideString; const ToDD : WideString; const ToMM : WideString; const ToYY : WideString): TAIMSPendNotifList;		
TAIMSPendNotifList = class(TRemotable)		
PendNotifList	TPendNtfList	= array of TAIMSCrewLogltn ;
PendNotifCount	Integer	
ErrorExplanation	WideString	

11. Remove Crew Notification Indicator on a certain date

The function **RemoveCrewNotificationIndicator** removes all notification indicators for a specific crew member on a specific day and returns a True/False result

function RemoveCrewNotificationIndicator(const UN : WideString; const PSW : WideString; const ID : WideString; const ForDD : WideString; For Date Day const ForMM : WideString; For Date Month const ForYY : WideString; For Date Year const OnDD : WideString; On Date Day const OnMM : WideString; On Date Month const OnYY : WideString; On Date Year const OnTimeMins : WideString; On Time (in minutes) const OnTimeMSecs : WideString; On Time (in milliseconds) var Error : WideString): Boolean;		
---	--	--

OnDD, OnMM, OnYY

Those fields requires the Day, Month and Year accordingly for the removal of the notification indicators On that day

OnTimeMins

This field requires the time in minutes for the removal of the notification indicators on that day, i.e. all notifications from that minute and backwards until the start of the day will be removed (i.e. each day has a total of 1440 minutes = 24hrs * 60mins).

OnTimeMSecs

This field requires the time in milliseconds for the removal of the notification indicators on that day, i.e. all notifications from that millisecond and backwards until the start of the day will be removed (i.e. each day has a total of 86400000 milliseconds = 24hrs * 60mins* 60secs * 1000 milliseconds).

In case you need the accuracy for which notifications you have to remove on that day, you must enter OnDD, OnMM,OnYY, OnTimeMins, OnTimeMSecs parameters to the Web Service. However, if this is not required, then by passing zero to all parameters, the system will remove all notifications on that day as it used to.

12. List crew personal information (inc Quals) for a specific crew or a group for a given period

The function **GetCrewList** returns the most basic information (e.g. ID, Name, Address, State etc.) for a specific crew member or a group of crew members for a given period.

<pre>function GetCrewList(const UN : WideString; const PSW : WideString; const ID : Integer; const PrimaryQualify : Boolean; const FmDD : WideString; const FmMM : WideString; const FmYY : WideString; const ToDD : WideString; const ToMM : WideString; const ToYY : WideString; const BaseStr : WideString; const ACStr : WideString; const PosStr : WideString) : TAIMSGetCrewList;</pre>		
TAIMSGetCrewList = class(TRemotable)		
GetCrewListCount	Integer	
GetCrewList	TGetCrewList	= array of TAIMSGetCrewltn
ErrorExplanation	WideString	

13. List Personal Crew Schedule information for a specific crew for a given period

The function **CrewMemberRosterDetailsForPeriod** returns the detailed crew schedule information (e.g Day,Carrier,Flt,Dep,Arr,STD,STA etc.) for a specific crew member for a given period.

<pre>function CrewMemberRosterDetailsForPeriod(const UN : WideString; const PSW : WideString; const ID : Integer; const FmDD : WideString; const FmMM : WideString; const FmYY : WideString; const ToDD : WideString; const ToMM : WideString; const ToYY : WideString): TAIMSCrewRostDetailList;</pre>		
TAIMSCrewRostDetailList = class(TRemotable)		
CrewRostCount	Integer	
CrewRostList	TCrewRostList	= array of TAIMSCrewRostltn
ErrorExplanation	WideString	

Refer to Field List 23

14. List Crew Qualifications for a specific crew or a group for a given period

The function 'FetchCrewQuals' returns the most basic information (e.g. ID, Name, Address, State etc.) along with the qualifications saved for a specific crew member or a group of crew members for a given period and accepts the following types of parameters:

- a) The date period on which the requested crew member(s) is/are qualified
- b) CrewID: The Crew member ID
- c) PrimaryQualify: Request for Primary/Secondary qualifications in the period above
- d) GetAllQualsInPeriod: Request to fetch all the qualifications in the period above.

When the (b) parameter is equal to zero the system will try to fetch all the crew members that are qualified in the date period requested (a), otherwise the system will fetch all the qualified members in that period according to the (c) parameter True=Primary, False=Secondary qualified. The (d) parameter will set if the returned list (with the qualifications for each crew member) will contain only the qualification records that fall in the requested period (=false) or all (=true).

function FetchCrewQuals(const UN : WideString; const PSW : WideString; const FmDD : WideString; const FmMM : WideString; const FmYYYY : WideString; const ToDD : WideString; const ToMM : WideString; const ToYYYY : WideString; const CrewID : Integer; const PrimaryQualify : Boolean; const GetAllQualsInPeriod : Boolean): TAIMSCrewQualsList;		
TAIMSCrewQualsList = class(TRemotable)		
TotalCrew	Integer	Qualified Members found on that period
CrewQualList	TAIMSCrewQualsLst	= array of TAIMSCrewQualsRec
ErrorExplanation	WideString	Message if an error occurs

15. Record fact that crew has checked in and provide details of the duty

The method **CrewCheckIn** is expected to return the next possible duty for a crew member for which he/she can check-in (for the current time) when the method is called with the 'Confirmed' argument equal to False. Otherwise it performs the check-in upon calling the method with 'Confirmed' argument equals to True.

function CrewCheckIn(const UN: WideString; const PSW: WideString; const ID: Integer; const Confirmed: Boolean): TAIMSCrewCheckIn;		
TAIMSCrewCheckIn = class(TRemotable)		
ErrorExplanation	WideString	Message if an error occurs
Status	Integer	Error code
Confirmed	Boolean	False=try Checkin / True=perform Checkin
Id	Integer	Crewmember Identification number
Name	WideString	Crewmember name
ShortName	WideString	Crew three letter code
Base	WideString	Qualification Main Base
Ac	WideString	Aircraft Type Qualification
Pos	WideString	Qualification Position
FlightNo	WideString	Flight number
FlightDate	WideString	Date on which the flight operated
FlightDep	WideString	Departure station
FlightArr	WideString	Arrival station
FlightStd	WideString	Scheduled Time of Departure
FlightSta	WideString	Scheduled Time of Arrival

16. Records fact that crew has checked out and provides details of the duty

The method **CrewCheckOut** is expected to return the next possible duty for a crew member for which he/she can check-out (for the current time) when the method is called with the 'Confirmed' argument equal to False. Otherwise it performs the check-out upon calling the method with 'Confirmed' argument equals to True.

function CrewCheckOut(const UN: WideString; const PSW: WideString; const ID: Integer; const Confirmed: Boolean): TAIMSCrewCheckOut;		
TAIMSCrewCheckOut = class(TRemotable)		
ErrorExplanation	WideString	Message if an error occurs
Status	Integer	Error code
Confirmed	Boolean	False=try Checkout / True=perform Checkout
Id	Integer	Crewmember Identification number
Name	WideString	Crewmember name
ShortName	WideString	Crew three letter code
Base	WideString	Qualification Main Base
Ac	WideString	Aircraft Type Qualification
Pos	WideString	Qualification Position
FlightNo	WideString	Flight number
FlightDate	WideString	Date on which the flight operated
FlightDep	WideString	Departure station
FlightArr	WideString	Arrival station
FlightStd	WideString	Scheduled Time of Departure
FlightSta	WideString	Scheduled Time of Arrival

17. List of Crew Members and their Max FDP values on a flight leg

The function **FetchCrewMaxFDP** returns the crew members found on a specific 'Flight' and obtain their Max FDP assigned on the flight duty.

function FetchCrewMaxFDP (const UN : WideString; const PSW : WideString; const DD : WideString; const MM : WideString; const YY : WideString; const FlightNo : WideString; (= CarrierCode+FlightNo+LegCD) const Dep : WideString): TAIMSGetCrewMaxFDP		
TAIMSGetCrewMaxFDP = class(TRemotable)		
FlightDD	WideString	Day on which the flight operated
FlightMM	WideString	Month in which the flight operated
FlightYY	WideString	Year in which the flight operated
FlightCarrier	WideString	Carrier Code
FlightNo	Integer	Flight number
FlightLegCD	WideString	Leg code flight operation suffix
FlightDesc	WideString	FlightCarrierCode + FFlightNo + FFlightLegCD
FlightDep	WideString	Flight Departure station
Count	Integer	Members found on that day
CrewMaxFDP	TCrewMaxFDPList	TCrewMaxFDPList = array of TAIMSCrewMaxFDP
ErrorExplanation	WideString	Message if an error occurs

18. List of Crew Messages

The method **FetchCrewMessages** is expected to return all the messages sent from crewing to crew regarding a specific crew member ID for the current day. The “Dictionary” method has been used in order to separate the messages returned by the method as it is compatible with portable devices using any mobile software like iOS or Android etc.

```
Function FetchCrewMessages(UN,PSW: String; ID: Integer; const DictKey: Integer): WideString;
```

The following structure is used for each message returned by the web service as shown bellow:

```
<dict><key>Must Acknowledge</key><integer>message id number</integer>  
<key>Message Text</key><string>your message</string></dict>
```

The “DictKey” field is used when an acknowledgment of a specific message is required and it stands for the message ID number.

If the ‘message id number’ value of the Integer part in the message “KEY” has a value greater than Zero (0), then the specific message needs acknowledgment. Otherwise, messages with values “-1” do not require any acknowledgment.

Flight Related

19. Flight Details for all flights of a specific day

The function **FlightDetailsForFlightLegOnDay** returns a list of Flight Details for a flight leg of a specific day

function FlightDetailsForFlightLegOnDay(const UN : WideString; const PSW : WideString; const DD : WideString; const MM : WideString; const YY : WideString): TAIMSFLightsList;		
TAIMSFLightsList = class(TRemotable)		
FlightCount:	Integer	
FlightList:	TLegslist	= array of TAIMSFlight ;
Ferrorexplanation	WideString	Message if an error occurs

20. Flight Details for all flights in the selected date/time range

The function **FlightDetailsForPeriod** returns a list of Flight Details for a flight leg of a specific period.

function FlightDetailsForPeriod (const UN : WideString; const PSW : WideString; const FromDD : WideString; const FromMMonth : WideString; const FromYYYY : WideString; const FromHH : WideString; const FromMMin : WideString; const ToDD : WideString; const ToMMonth : WideString; const ToYYYY : WideString; const ToHH : WideString; const ToMMin : WideString): TAIMSFLightsList;		
TAIMSFLightsList = class(TRemotable)		
FlightCount:	Integer	
FlightList:	TLegslist	= array of TAIMSFlight ;
Ferrorexplanation	WideString	Message if an error occurs

21. Flight details for specific flights for the selected date/time range

The function **FlightDetails** returns a list of Flight Details for flight legs of a specific period. The method now includes a plethora of parameters i.e. Dep station, Arr station, Aircraft type etc. in order to a) reduce the amount of results returned in the requested period/parameters and consequently decrease the processing time as well and b) through the provision of filtering these parameters, allow Users to request for more concise and specific information.

function FlightDetails (const UN	: WideString;	Authentication UserName
const PSW	: WideString;	Authentication Password
const FromDD	: WideString;	Request period Start Day
const FromMMonth	: WideString;	Request period Start Month
const FromYYYY	: WideString;	Request period Start Year
const FromHH	: WideString;	Request period Start Hour
const FromMMin	: WideString;	Request period Start Minute
const ToDD	: WideString;	Request period End Day
const ToMMonth	: WideString;	Request period End Month
const ToYYYY	: WideString;	Request period End Year
const ToHH	: WideString;	Request period End Hour
const ToMMin	: WideString;	Request period End Minute
const FlightCarrier	: WideString;	Carrier Code
const FlightNum	: Integer;	Flight number
const FlightLegCd	: Char;	Leg code flight operation suffix
const Dep	: WideString;	Request Departure station
const Arr	: WideString;	Request Arrival station
const ACType	: WideString;	Request Aircraft Type
const DepArrMode*	: Byte;) : TAIMSFLightsList;
*DepArrMode : 1=Collect departing legs / 2=Collect arriving legs in period above		
TAIMSFLightsList = class(TRemotable)		
FlightCount:	Integer	
FlightList:	TLegslist	= array of TAIMSFlight ;
Ferrorexplanation	WideString	Message if an error occurs

22. Flight Schedule changes for a given period

The function **FlightScheduleChangesForPeriod** returns a list of changes made to flights within a specific dates range and time range.

function FlightScheduleChangesForPeriod(const UN	: WideString;	
const PSW	: WideString;	
const FromDD	: WideString;	
const FromMMonth	: WideString;	
const FromYYYY	: WideString;	
const FromHH	: WideString;	
const FromMMin	: WideString;	
const ToDD	: WideString;	
const ToMMonth	: WideString;	
const ToYYYY	: WideString;	
const ToHH	: WideString;	
const ToMMin	: WideString;	TAIMsFlightsLogList;
TAIMsFlightsLogList = class(TRemotable)		
LogList:	TLogList	
ErrorExplanation:	WideString	= array of TAIMsFlightLog ;
FlightLogCount:	Integer	

23. Flight Schedule Modification Log for a given period

The function **FlightScheduleModificationLog** returns a list of changes made to the flight schedule for a specified period and time range.

Period FOR which the changes were made

Allows a user to search a period for which an actual modification has been made.

(e.g. The ETD of Flight 123 was changed from 24:30 to 01:30 on the following day “01/01/2014” – search for this date)

Period ON which the changes were made

Allows a user to search a period on which the change was made.

(e.g. Search a period when a schedule was physically changed – above change to schedule was made on 30/12/2013 – search for this date).

Note: If the flight leg data has been removed (i.e. flight ‘Deleted’) then only flight leg information recorded to the log file will be viewed e.g. Date of the flight, Flight number, Departure station etc.

```
function FlightScheduleModificationLog(const UN      : WideString;
                                     const PSW      : WideString;
                                     const ForBegDD   : WideString; ForDate Start Day
                                     const ForBegMM   : WideString; ForDate Start Month
                                     const ForBegYYYY : WideString; ForDate Start Year
                                     const ForEndDD   : WideString; ForDate End Day
                                     const ForEndMM   : WideString; ForDate End Month
                                     const ForEndYYYY : WideString; ForDate End Year
                                     const OnBegDD    : WideString; OnDate Start Day
                                     const OnBegMM    : WideString; OnDate Start Month
                                     const OnBegYYYY  : WideString; OnDate Start Year
                                     const OnBegHHrs  : WideString; OnDate Start Hour
                                     const OnBegMMin   : WideString; OnDate Start Minute
                                     const OnEndDD     : WideString; OnDate End Day
                                     const OnEndMM    : WideString; OnDate End Month
                                     const OnEndYYYY  : WideString; OnDate End Year
                                     const OnEndHHrs  : WideString; OnDate End Hour
                                     const OnEndMMin   : WideString; OnDate End Minute); TAimsFlightsSchedModLogList;
```

TAimsFlightsSchedModLogList = class(TRemotable)

FltsSchedModLogList : TFltsSchedModLogList = array of [TAimsFltsSchedModLogItem](#);

FltsSchedModLogCount: Integer Changes found on the requested period

ErrorExplanation : WideString Message if an error occurs

24. Upload Journey Log data for a specific flight leg

The function UpdateVRData attempts to insert Journey Log data to the flight schedule database related files with details that applied to the TAIMSVRData parameter object e.g. FlightDay, FlightMonth, FlightYear, FlightNum, LandsTkof details etc. On success it will return a True value, otherwise it returns False and the "Error" explanation. It also revert back any changes already performed to the related files for the requested flight leg.

```
function UpdateVRData (const UN      : WideString;
                      const PSW     : WideString;
                      const VRData   : TAIMSVRData;
                      var  Error     : WideString >> Message if an error occurs): Boolean;
On success it will return a True value, otherwise it returns False and the "Error" explanation.
```

TAIMSVRData = class(TRemotable)		
FlightDay	WideString	Day on which the flight operated
FlightMonth	WideString	Month in which the flight operated
FlightYear	WideString	Year in which the flight operated
FlightDep	WideString	Departure station
FlightCarrier	WideString	Carrier Code
FlightNum	WideString	Flight number
FlightLegCd	WideString	Leg code flight operation suffix
ActualTimes	TAIMSActualTimesRec	VR Actual times of flight
DelayTimes	TAIMSDelayTimesRec	VR Delay times of flight
LandsTkof	TAIMSLandsTkofItems	= array of TAIMSLandsTkofRec Up to the maximum flight assigned members
Facilities	TAIMSFacilitiesItems	= array of TAIMSFacilitiesRec Up to xx items for Facilities entries
Pasengers	TAIMSPasengersItems	= array of TAIMSPasengersRec Up to xx items for Pasengers entries
FuelItems	TAIMSFuelCargoItems	= array of TAIMSFuelCargoRec Up to 20 items for Fuel entries
CargoItems	TAIMSFuelCargoItems	= array of TAIMSFuelCargoRec Up to 10 items for Cargo entries
Suppliers	TAIMSSupplierItems	= array of TAIMSSupplierRec Up to 6 items for Uplift / Slip entries
AdditionalInfo	TAIMSAdditionalInfo	Additional fields regarding the flight operated

25. List of Aircraft Routes with its flight legs for a given period

The function **FetchAircraftRouteDataAndSectors** returns detailed information for all aircraft routes, along with sector details for a given period. .

function FetchAircraftRouteDataAndSectors (const UN: WideString; const PSW: WideString; const FromDD: WideString; const FromMM: WideString; const FromYYYY: WideString; const ToDD: WideString; const ToMM: WideString; const ToYYYY: WideString): TAIMSGetRouteAndSectorList;		
TAIMSGetRouteAndSectorList = class(TRemotable)		
AIMSARouteList	TAIMSARouteList	= array of TAIMSARoute
rCount	Integer	
rErrorExplanation	WideString	

26. Upload various leg information for a specific flight leg

The function UpdateLegInfo attempts to insert various information to the flight schedule database related files with details that applied to the TAIMSUpdLegInfo parameter object e.g. FlightDay, FlightMonth, FlightYear, FlightNum, DEP/ARR - Gate/Stand details etc. On success it will return a True value, otherwise it returns False and the "Error" explanation. It also revert back any changes already performed to the related files for the requested flight leg.

```
function UpdateLegInfo(const UN : WideString;
                      const PSW: WideString;
                      Const UpdLegInfo: TAIMSUpdLegInfo;
                      var Error : WideString >>>> Message if an error occurs): Boolean;
```

TAIMSUpdLegInfo = class(TRemotable)		
FlightDay	WideString	Day on which the flight operated
FlightMonth	WideString	Month in which the flight operated
FlightYear	WideString	Year in which the flight operated
FlightDep	WideString	Departure station
FlightCarrier	WideString	Carrier Code
FlightNum	WideString	Flight number
FlightLegCd	WideString	Leg code flight operation suffix
DepStand	WideString	Departure Stand of the flight
ArrStand	WideString	Arrival Stand of the flight
DepGate	WideString	Departure Gate of the flight
ArrGate	WideString	Arrival Gate of the flight

Miscellaneous Web Services

27. List of Aircraft

The function **FetchAircraft** returns detailed information for all aircraft.

function FetchAircraft(const UN: WideString; const PSW: WideString): TAIMSGetAircraftList;		
TAIMSGetAircraftList = class(TRemotable)		
cAIMSAircraftList	TAIMSAircraftCList	= array of TAIMSAircraft
cCount	Integer	
cErrorExplanation	WideString	

28. List of Aircraft Types

The function **FetchACType** returns detailed information for all aircraft types.

function FetchACType(const UN: WideString; const PSW: WideString): TAIMSGetACTypeList;		
TAIMSGetACTypeList = class(TRemotable)		
cAIMSACTypeList	TAIMSACTypeList	= array of TAIMSACType
cCount	Integer	
cErrorExplanation	WideString	

29. List of Countries

The function **FetchCountries** returns detailed information for a specific country code. If the 'CountryCode' parameter is blank then the function will return a list of all the countries.

function FetchCountries(const UN: WideString; const PSW: WideString; const CountryCode: WideString): TAIMSGetCountryList;		
TAIMSGetCountryList= class(TRemotable)		
cCountryList	TAIMSCountryList	= array of TAIMSCountry
cCount	Integer	
cErrorExplanation	WideString	

30. List of Airports

The function **FetchAirports** returns detailed information for all airports.

function FetchAirports(const UN: WideString; const PSW: WideString): TAIMSGetAirportList;		
TAIMSGetAirportList = class(TRemotable)		
cAIMSAirportList	TAIMSAirportList	= array of TAIMSAirport
cCount	Integer	
cErrorExplanation	WideString	

31. List of Charterers

The function **FetchCharterers** returns detailed information for all charterers.

function FetchCharterers(const UN: WideString; const PSW: WideString): TAIMSChartRecList;		
TAIMSChartRecList = class(TRemotable)		
ChartererList	TChartArray	= array of TAIMSChartRec
rCount	Integer	
ErrorExplanation	WideString	

32. Time difference between UTC/Local for a station for a given date and time

The function **GreenwichMeanTimeDifferenceInMins** returns the difference in minutes (either positive or negative) between UTC and a specific station for a given date (onYear, onMonth, onDate) and time (onHours, onMins), and on any case of failure the field 'Error' will hold the reason why the method failed e.g. field 'onYear' is empty or is not numeric.

Additionally, the user has the ability to pass to the web service the way that should calculate the time difference by setting one of the following values to the field 'TypeOfTime':

- 0 = requested time is as the time used in AIMS database files (7.1 AIMS Control Information | All Times In: UTC / Local Base)
- 1 = requested time is in UTC.
- 2 = requested time is in Local time.

function GreenwichMeanTimeDifferenceInMins(const UN	: WideString;
const PSW	: WideString;
const onBase	: WideString;
const onYear	: WideString;
const onMonth	: WideString;
const onDate	: WideString;
const onHours	: WideString;
const onMins	: WideString;
const TypeOfTime	: WideString;
var Error	: WideString): Smallint;

Field Lists

TAIMSMember = class(TRemotable)		
id	Integer	Crewmember Identification number
name_	WideString	Crewmember name
lastname	WideString	Crewmember surname (*as shown within 8.8 Crew Documents tab, Passport names)
firstname	WideString	Crewmember first name (*)
shortname	WideString	Crew three letter code
base	WideString	Qualification Main Base
ac	WideString	Aircraft Type Qualification
pos	WideString	Qualification Position
qualcat	Byte	Qualification category
trnduty	WideString	Duty in crew route
request	WideString	If crew route is a member's request
notif	WideString	If member is notified for crew route
pax	WideString	Indicates if member is pax on this leg
crte	WideString	Crew route number member belongs to
crsday	Smallint	Start day of crew route
crsdaydd	WideString	Day of start day of crew route
crsdaymm	WideString	Month of start day of crew route
crsdayyy	WideString	Year of start day of crew route

TAIMSCrewLogltn = class(TRemotable)		
LogForDay	WideString	Date FOR which change was made
LogMessage	WideString	Displays pending notifications
LogByUser	WideString	Logged by User X
LogOnDate	WideString	Date ON which change was made
LogOnTime	WideString	Time that change was made
LogOnDayTime	WideString	Day & Time change was made

TAIMSunfinRteltem = class(TRemotable)		
UnFinRoute	WideString	Unfinalised Crew Route number
UnFinDate	WideString	Date of Unfinalised Crew Route (DD/MM/YYYY)
UnFinDD	WideString	Day of Unfinalised Crew Route
UnFinMM	WideString	Month of Unfinalised Crew Route
UnFinYY	WideString	Year of Unfinalised Crew Route
ErrorExplanation	WideString	* Informative message describing any issues that may have occurred during the retrieval of information

TAIMSCrwMissTrnIndltem = class(TRemotable)		
7F_ID	Integer	Missing Training Indicator for Crew ID 123
MissTrnInd_NAME	WideString	Missing Training Indicator for Crew Name
MissTrnInd_BASE	WideString	
MissTrnInd_AC	WideString	
MissTrnInd_POS	WideString	
MissTrnInd_Date	WideString	
MissTrnInd_DD	WideString	
MissTrnInd_MM	WideString	
MissTrnInd_YY	WideString	
ErrorExplanation	WideString	* Informative message describing any issues that may have occurred during the retrieval of information

TAIMSGetCrewItm = class(TRemotable)		
Id	WideString	Crewmember Identification number
ShortName	WideString	Crew three letter code
CrewName	WideString	Crewmember name
Location	WideString	Qualification Main Base
Quals	WideString	Base, AC, Position
Sex	WideString	M/F
Marital	WideString	Marital Status
Address	WideString	
City	WideString	
State	WideString	
Zip	WideString	
NationalityCode	WideString	
Nationality	WideString	
ContactCell	WideString	
ContactTel	WideString	
Email	WideString	
NextKinName	WideString	Next of KIN name
NextKinRelsn	WideString	Relation to next of KIN
NextKinContact	WideString	Next of KIN phone number
EmploymentDate	WideString	Commencement of employment date
Language1	WideString	First Language qualification code
Language2	WideString	Second Language qualification code
LicenceNum	WideString	License number (8.8 Crew Documents)

TAIMSQualRec = class(TRemotable)		
QualAc	WideString	Aircraft Type Qualification
QualBase	WideString	Qualification Main Base
QualPos	WideString	Qualification Position
QualBDay	WideString	Qualification Start Date (YYYYMMDD)
QualEDay	WideString	Qualification End Date (YYYYMMDD)

TAIMSCrewInfo = class(TRemotable)		
CrewID	WideString	Crewmember Identification number
CrewName	WideString	Crewmember name
Crew3LC	WideString	Crew three letter code
Gender	WideString	M/F
CrewQual	TAIMSQualItems	= array of TAIMSQualRec;
Address1	WideString	First line of physical address
Address2	WideString	Second line of physical address
Address3	WideString	Third line of physical address
City	WideString	
State	WideString	
ZipCode	WideString	
Residence	WideString	Country of residence
Telephone	WideString	
Fax	WideString	
CellPhone	WideString	
Email	WideString	
Email2	WideString	
KinName	WideString	Next of KIN name
Relation	WideString	Relation to next of KIN
Contact	WideString	Next of KIN phone number
Marital	WideString	Marital Status (e.g. Single, Married, divorced etc.)
EmployBeg	WideString	Employment begin date
EmployEnd	WideString	Employment end date
EndReason	WideString	Reason employment was ceased
Mailbox	WideString	Mailbox number
PayrollNm	WideString	Payroll number
SecNumber	WideString	Social Security number

TAIMSCrwCtgsRec = class(TRemotable)		
CrewCatIdx	Smallint	Crew Category Index
CrewCatBDate	WideString	Crew Category begin date
CrewCatEDate	WideString	Crew Category end date

TAIMSQualsRec2 = class(TRemotable)		
QualBase	WideString	Qualification Main Base
QualAc	WideString	Aircraft Type Qualification
QualPos	WideString	Qualification Position
RosterGroup	Smallint	Qualification Roster Group
QualBDay	WideString	Qualification Start Date (YYYYMMDD)
QualEDay	WideString	Qualification End Date (YYYYMMDD)
PrimaryQual	Boolean	True = Primary Qualification
NotToBeRoster	Boolean	Qualification field 'Not to be Roster flights of this A/C'

TAIMSCrewInfo2 = class(TRemotable)		
CrewID	WideString	Crewmember Identification number
CrewName	WideString	Crewmember name
Gender	WideString	M/F
EmployBeg	WideString	Employment begin date
ExemptSBY	Boolean	Stand by duties exempted
LeaveCateg	WideString	Leave Category
UnderTrain	Boolean	Crew under training
CrewCategs	TAIMSCrwCtgsItems	= array of TAIMSCrwCtgsRec
CrewQuals	TAIMSQualsItems2	= array of TAIMSQualsRec2;

TAIMSCrewRostItm = class(TRemotable)		
Day	WideString	Day of operation
Carrier	WideString	Carrier code
Flt	WideString	Flight number
Legcd	WideString	Leg code flight operation suffix
Dep	WideString	Departure Station
Arr	WideString	Arrival Station
Crew base	WideString	Crew base for this day, include rule about change base after 48 hours out of base.
STD	WideString	
STDLocal	WideString	Schedule time of departure in Local time
STDLocalBase	WideString	Schedule time of departure in Local Crew Base time for this day
STA	WideString	
STALocal	WideString	Schedule time of arrival in Local time
STALocalBase	WideString	Schedule time of arrival in Local Crew Base time for this day
ATD	WideString	
ATDLocal	WideString	Actual time of departure in Local time
ATDLocalBase	WideString	Actual time of departure in Local Crew Base time for this day
ATA	WideString	
ATALocal	WideString	Actual time of arrival in Local time
ATALocalBase	WideString	Actual time of arrival in Local Crew Base time for this day
GDBEG	WideString	Duty period start time
GDBEGLocal	WideString	Duty period start time in Local time
GDBEGLocalBase	WideString	Duty period start time in Local Crew Base time for this day
GDEND	WideString	Duty period end time
GDENDLocal	WideString	Duty period end time in Local time
GDENDLocalBase	WideString	Duty period end time in Local Crew Base time for this day
PAX	WideString	Operated as PAX
CROUTE	WideString	Crew Route number
CDATE	WideString	Crew Route start date

= class(TRemotable)		
Rank	WideString	Qualification Position
StaffNum	WideString	Crewmember Identification number
CrewName	WideString	Crewmember name
FDPStart	WideString	FDP start time
MaxFDP	WideString	Max FDP time
FDPend	WideString	FDP end time
CrewRoute	WideString	Crew route number member belongs to
CRouteDate	WideString	Start day of Crew route

TAIMSFlight = class(TRemotable)		
FlightDD	WideString	Day on which the flight operated
FlightMM	WideString	Month in which the flight operated
FlightYY	WideString	Year in which the flight operated
FlightDate	WideString	Date on which the flight operated
FlightCarrier	WideString	Carrier Code
FlightNo	Integer	Flight number
FlightDep	WideString	Departure station
FlightArr	WideString	Arrival station
FlightStd	WideString	Scheduled Time of Departure
FLightSta	WideString	Scheduled Time of Arrival
FlightEtd	WideString	Estimated Time of Departure
FlightEta	WideString	Estimated Time of Arrival
FlightAtd	WideString	Actual Time of Departure
FlightAta	WideString	Actual Time of Arrival
FlightReg	WideString	Flight Registration
FlightAcType	WideString	Flight Aircraft Type (e.g. 757, 320)
FlightType	WideString	Flight Service Type (e.g. Scheduled, Charter etc.)
FlightBlkTime	WideString	Block time of flight
FlightLegCD	WideString	Leg code flight operation suffix
Ferrorexplanation	WideString	Message if an error occurs
FlightDesc	WideString	FlightCarrierCode + FFlightNo + FFlightLegCD
FlightStatus	WideString	Descriptive message for the flight: Arrived / In flight / Not yet departed
FlightTKOFF	WideString	Actual Take Off Time
FlightTDOWN	WideString	Actual Touch Down Time
FlightDepDelays	TAIMSFlightDelays	=array of TAIMSDelaysRec (Up to 8 departure delays)
FlightArrDelays	TAIMSFlightDelays	=array of TAIMSDelaysRec (Up to 2 arrival delays)
FlightNoOfPax	Integer	Number of Passengers in the Flight (2016B2 & onwards)
FlightAssocCrewRtes	TAIMSAssocCwRtes	=array of TAIMSAssocCwRtesltn List which contains the crew routes and starting dates where the flight leg is used.

TAimsFlightLogDetails = class(TRemotable)		
LogForDay	WideString	Changed For day
LogDep	WideString	Flight Departure
LogCarrier	WideString	Carrier
LogFlt	Integer	Flight number
LogLegCd	WideString	Leg code flight operation suffix
LogOnDay	WideString	Changed On day
LogOnTime	WideString	At XX:XX
LogDayTime	WideString	Day + XX:XX
LogDescription	TLogDetailDescription	= array of TAimsFLogDetailsDescription;
LogChangedBy	WideString	User who made the change
LogDescrCount	Integer	# of lines found in description

TAimsFlightLog = class(TRemotable)		
LogDetailsList	TLogDetailsList	= array of TAimsFlightLogDetails;
ErrorExplanation	WideString	*
LogDetailsCount	Integer	# of lines found in the details list

TAimsFltsSchedModLogItem = class(TRemotable)

FltsSchedModLog_Day	: WideString;	Day of operation
FltsSchedModLog_Flt	: Integer;	Flight number
FltsSchedModLog_Dep	: WideString;	Departure station
FltsSchedModLog_Arr	: WideString;	Arrival station
FltsSchedModLog_Carrier	: WideString;	Carrier code
FltsSchedModLog_LegCd	: Char	Leg code flight operation suffix
FltsSchedModLog_UTC_STD	: WideString;	Scheduled Time of Departure in UTC
FltsSchedModLog_UTC_STA	: WideString;	Scheduled Time of Arrival in UTC
FltsSchedModLog_UTC_ETD	: WideString;	Estimated Time of Departure in UTC
FltsSchedModLog_UTC_ETA	: WideString;	Estimated Time of Arrival in UTC
FltsSchedModLog_UTC_ATD	: WideString;	Actual Time of Departure in UTC
FltsSchedModLog_UTC_ATA	: WideString;	Actual Time of Arrival in UTC
FltsSchedModLog_UTC_TOFF	: WideString;	Actual Take Off Time in UTC
FltsSchedModLog_UTC_TDWN	: WideString;	Actual Touch Down Time in UTC
FltsSchedModLog_LOCSTN_STD	: WideString;	Scheduled Time of Departure in Local Station
FltsSchedModLog_LOCSTN_STA	: WideString;	Scheduled Time of Arrival in Local Station
FltsSchedModLog_LOCSTN_ETD	: WideString;	Estimated Time of Departure in Local Station
FltsSchedModLog_LOCSTN_ETA	: WideString;	Estimated Time of Arrival in Local Station
FltsSchedModLog_LOCSTN_ATD	: WideString;	Actual Time of Departure in Local Station
FltsSchedModLog_LOCSTN_ATA	: WideString;	Actual Time of Arrival in Local Station
FltsSchedModLog_LOCSTN_TOFF	: WideString;	Actual Take Off Time in Local Station
FltsSchedModLog_LOCSTN_TDWN	: WideString;	Actual Touch Down Time in Local Station
FltsSchedModLog_Status	: WideString;	Descriptive message for the flight: Arrived / Deleted etc.
FltsSchedModLog_FlightType	: WideString;	Flight Service Type (e.g. Scheduled, Charter etc.)
FltsSchedModLog_NoOfPax	: Integer;	Number of Passengers in the Flight
FltsSchedModLog_ExpPassngr	: Integer;	Number of Expected Passengers in the Flight
FltsSchedModLog_DlyCode1	: WideString;	Departure Flight Delay Code No1
FltsSchedModLog_DlyTime1	: WideString;	Departure Flight Delay Time (HH:MM) No1
FltsSchedModLog_DlyCode2	: WideString;	Departure Flight Delay Code No2
FltsSchedModLog_DlyTime2	: WideString;	Departure Flight Delay Time (HH:MM) No2
FltsSchedModLog_DlyCode3	: WideString;	Departure Flight Delay Code No3
FltsSchedModLog_DlyTime3	: WideString;	Departure Flight Delay Time (HH:MM) No3
FltsSchedModLog_DlyCode4	: WideString;	Departure Flight Delay Code No4
FltsSchedModLog_DlyTime4	: WideString;	Departure Flight Delay Time (HH:MM) No4
FltsSchedModLog_ArrDlyCode1	: WideString;	Arrival Flight Delay Code No1
FltsSchedModLog_ArrDlyTime1	: WideString;	Arrival Flight Delay Time (HH:MM) No1
FltsSchedModLog_ArrDlyCode2	: WideString;	Arrival Flight Delay Code No2
FltsSchedModLog_ArrDlyTime2	: WideString;	Arrival Flight Delay Time (HH:MM) No2
FltsSchedModLog_DepGate	: WideString;	Departure Gate
FltsSchedModLog_ArrGate	: WideString;	Arrival Gate
FltsSchedModLog_DepStand	: WideString;	Departure Stand
FltsSchedModLog_ArrStand	: WideString;	Arrival Stand
FltsSchedModLog_AcType	: WideString;	Flight Aircraft Type (e.g. 757, 320)
FltsSchedModLog_AcOwner	: WideString;	Flight Aircraft Type Owner Code
FltsSchedModLog_AcReg	: WideString;	Flight Registration
FltsSchedModLog_OnDay	: WideString;	Changed On day
FltsSchedModLog_OnTime	: WideString;	At XX:XX

TAimsFLogDetailsDescription = class(TRemotable)		
LogDescription	WideString	Description of modification log (for changes made to roster scheduled data).

TAIMSActualTimesRec = class(TRemotable)		
ATD_HH	WideString	Blocks off hour(s)
ATD_MM	WideString	Blocks off minute(s)
TKOF_HH	WideString	Take off hour(s)
TKOF_MM	WideString	Take off minute(s)
TDOWN_HH	WideString	Touch down hour(s)
TDOWN_MM	WideString	Touch down minute(s)
ATA_HH	WideString	Blocks on hour(s)
ATA_MM	WideString	Blocks on minute(s)

TAIMSDelayTimesRec = class(TRemotable)		
FDepDelayTime1_HH	WideString	1st Departure delay hour(s)
FDepDelayTime1_MM	WideString	1st Departure delay minute(s)
FDepDelayCode1	WideString	1st Departure delay code
FDepDelayTime2_HH	WideString	2nd Departure delay hour(s)
FDepDelayTime2_MM	WideString	2nd Departure delay minute(s)
FDepDelayCode2	WideString	2nd Departure delay code
FDepDelayTime3_HH	WideString	3rd Departure delay hour(s)
FDepDelayTime3_MM	WideString	3rd Departure delay minute(s)
FDepDelayCode3	WideString	3rd Departure delay code
FDepDelayTime4_HH	WideString	4th Departure delay hour(s)
FDepDelayTime4_MM	WideString	4th Departure delay minute(s)
FDepDelayCode4	WideString	4th Departure delay code
FArrDelayTime1_HH	WideString	1st Arrival delay hour(s)
FArrDelayTime1_MM	WideString	1st Arrival delay minute(s)
FArrDelayCode1	WideString	1st Arrival delay code
FArrDelayTime2_HH	WideString	2nd Arrival delay hour(s)
FArrDelayTime2_MM	WideString	2nd Arrival delay minute(s)
FArrDelayCode2	WideString	2nd Arrival delay code

TAIMSLandsTkofRec = class(TRemotable)		
If crew members assigned to the flight, these are the values for the Takeoff or Landing or both per crew member.		
TKFLND_ID	Integer	Crew member ID who performed Takeoff/Landing
TKOF_DAY	Boolean	True/False (If it was a day takeoff)
TKOF_NIGHT	Boolean	True/False (If it was a night takeoff)
TKOF_MAN	Boolean	True/False (If it was a manual takeoff)
TKOF_AUTO	Boolean	True/False (If it was a automatic takeoff)
LAND_DAY	Boolean	True/False (If it was a day landing)
LAND_NIGHT	Boolean	True/False (If it was a night landing)
LAND_MAN	Boolean	True/False (If it was a manual landing)
LAND_AUTO	Boolean	True/False (If it was a automatic landing)
CAT3_ACT	Boolean	True/False (If it was a actual Cat3)
CAT3_PRAC	Boolean	True/False (If it was a practice Cat3)
RHS	Boolean	True/False (If it was a RHS)

TAIMSFacilitiesRec = class(TRemotable)		
FacilDesc	WideString	Description of facility
FacilUsed	Boolean	True/False (if facility has been used)
FacilAmount	WideString	Value of facility (double values are valid too)

TAIMSPasengersRec = class(TRemotable)		
PasngrCode	WideString	Passenger code class (found on 7.6)
PasngrNum	Word	Number of passengers

TAIMSFuelCargoRec = class(TRemotable)		
FieldDescr	WideString	Heading description for Fuel or Cargo column entry respectively
Value	Double	Value for the above column entry
MeasureUnit	Byte	The measurement unit for the entry i.e. Kgs

TAIMSSupplierRec = class(TRemotable)		
If 'UPLIFT(Vol)' column entry exist in the FuelItems data entries these are the Uplift / Split figures		
Supl_AgentCompany	WideString	Fuel Agent name
Supl_Value	Integer	Uplift volume value
Supl_SlipNumber	Byte	Fuel slip value

TAIMSAdditionalInfo = class(TRemotable)		
Remarks	WideString	Remarks for the flight operated
ExtraFld1	WideString	The “ExtraFld1” is a generic field in order to be used for expanding the functionality of the web service based on customizations/modifications. (**)
ExtraFld2	WideString	The “ExtraFld2” is a generic field in order to be used for expanding the functionality of the web service based on customizations/modifications. (**)
		** (Note: This is an optional field and will be used on the web service requests for specific airlines).

TAIMSDelaysRec = class(TRemotable)		
DelayCode	WideString	Flight Delay Code
DelayTime	WideString	Flight Delay Time (HH:MM)

TAIMSAssocCrewRtesItm = class(TRemotable)		
AssocCrewRouteName	WideString	Crew route name i.e. 5555A
AssocCrewRouteDate	WideString	Crew route date i.e. 01/01/2017

TAIMSARoute = class(TRemotable)		
ARouteADate	WideString	AC Route + AC date
ARoute	WideString	AC Route
ADate	WideString	AC Date
AcSerialNumber	WideString	CERTAIN AIRLINES ONLY
AIMSSectorList	TAIMSSectorList	= array of TAIMSSector

TAIMSCrewQualsRec = class(TRemotable)		
CrewID	WideString	Crewmember Identification number
CrewName	WideString	Crewmember name
FirstName	WideString	Crewmember first name (*)
LastName	WideString	Crewmember surname (*)
Crew3LC	WideString	Crew three letter code
Gender	WideString	M/F
Address1	WideString	First line of physical address
Address2	WideString	Second line of physical address
Address3	WideString	Third line of physical address
City	WideString	
State	WideString	
ZipCode	WideString	
Residence	WideString	Country of residence
Telephone	WideString	
Fax	WideString	
CellPhone	WideString	
Email	WideString	
Email2	WideString	
KinName	WideString	Next of KIN name
Relation	WideString	Relation to next of KIN
Contact	WideString	Next of KIN phone number
Marital	WideString	Marital Status (e.g. Single, Married, divorced etc.)
EmployBeg	WideString	Employment begin date
EmployEnd	WideString	Employment end date
EndReason	WideString	Reason employment was ceased
Mailbox	WideString	Mailbox number
PayrollNm	WideString	Payroll number
SecNumber	WideString	Social Security number
CrewQuals	TAIMSCrewQualsItems	= array of TAIMSCrewQualsItem
ErrorExplanation	WideString	Message if an error occurs

(*) as shown within 8.8 | Crew Documents tab, Passport names

TAIMSCrewQualsItem = class(TRemotable)		
QualBase	WideString	Qualification Main Base
QualAc	WideString	Aircraft Type Qualification
QualPos	WideString	Qualification Position
RosterGroup	Smallint	Qualification Roster group
QualBDay	WideString	Qualification begin date
QualEDay	WideString	Qualification end date
PrimaryQual	Boolean	True = Primary qualification
NotToBeRoster	Boolean	Qualification field 'Not to be Roster flights of this A/C'

TAIMSSector = class(TRemotable)		
ARouteADate	WideString	AC Route + AC date
OperDate	WideString	Operation date
FltNo	Integer	Flight number
Dep	WideString	Departure station
LegCd	WideString	Leg code flight operation suffix
CarrierCode	WideString	Carrier Code
Arr	WideString	Arrival Station
CrewOnSectorArray	Array Of String	CERTAIN AIRLINES ONLY
Std	WideString	
Sta	WideString	
ArrDate	WideString	Arrival Date
ChartererCode	WideString	Charterer Code

TAIMSAircraft = class(TRemotable)		
cAcType	WideString	AC Type
cACSerialNumber	WideString	CERTAIN AIRLINES ONLY (CAO)
cACReg	WideString	AC Registration number
cACCcountry	WideString	Registered on the date of request
cACUsePound	Byte	(1=Kgs, 2=Pounds)
TAIMSACType = class(TRemotable)		
cAcTypeCode	WideString	AC Type
cDescription	WideString	AC Type Description
cCHCDescription	WideString	CERTAIN AIRLINES ONLY
TAIMSCountry = class(TRemotable)		
cCountryCode	WideString	Country code
cCountryName	WideString	Country name
TAIMSAirport = class(TRemotable)		
cAirportCode	WideString	Airport code
cAirportName	WideString	Airport name
cHeilideckHeight	Double	CERTAIN AIRLINES ONLY
cLatitude	Double	On date of request
cLongitude	Double	On date of request
cCountryCode	WideString	Country code
clsPossibleDiversionBit	Boolean	CERTAIN AIRLINES ONLY
clsPossibleFuelLocation	Boolean	On date of request - CAO
cLocationType	Integer	0: Airport, 1: Oil Rig or Ship – CAO
cAltitude	Integer	Airport Altitude
cRunwayLength	Integer	Runway length of airport
TAIMSCartRec = class(TRemotable)		
ChartererCode	WideString	Charter Code
ChartererDescription	WideString	Description of Charter
TAIMSAdditionalInfo = class(TRemotable)		
Remarks	WideString	Remarks for the flight operated
ExtraFld1	WideString	The "ExtraFld1" is a generic field in order to be used for expanding the functionality of the web service based on customizations/modifications. (Note: This is an optional field and will be included on the web service requests for specific airlines).