# System Interface Document

PROS

## PROS
## Group Revenue Management System

08 February 2018

PROS
3100 Main Street Suite 900,
Houston, TX 77002-9312 USA

**Product Information**

| | |
|---|---|
| Phone | 713-335-5100 |
| Email | info@pros.com |
| Web Address | http://www.pros.com |

## Change History

| VERSION | DATE | PERSON | REASON FOR CHANGE |
|---|---|---|---|
| Draft | 20 Jul 2005 | Sanjay Datta | First draft |
| 1.0 | 22 Jul 2005 | Carlos Hinke | Standard formatting for discussions with SITA |
| 1.1 | 15 Aug 2005 | Carlos Hinke | Add request loader format description per Sanjay Datta |
| 1.2 | 14 Sep 2005 | Tse Lee | Change the CCITIN and PnrGen interface formats. Modify the PnrLoad and Request Load XML format |
| | 16 Sep 2005 | Tse Lee | Add the reference load (IATA, CityPos, …) schema and examples |
| | 19 Sep 2005 | Tse Lee | Negotiation module documentation. Also correct the RequestLoad example |
| | 20 Sep 2005 | Tse Lee | Acceptance Module document and correct PnrGen PRO field |
| | 21 Sep 2005 | Tse Lee | Confirmation/Rejection document |
| | 28 Sep 2005 | Tse Lee | Confirmation/Rejection document and change the font style |
| | 29 Sep 2005 | Tse Lee | Add DirNum, RangeNum, and SegNum to the request load file |
| | 30 Sep 2005 | Tse Lee | Correct Office code example |
| | 3 Oct 2005 | Tse Lee | Correct > Fare and Special fare errors |
| | 4 Oct 2005 | Tse Lee | Correct impDate extra space from the Market Fare table |
| | 13 Oct 2005 | Tse Lee | Add the SubOptID for the break-up option for the negotiation module |
| | 14 Oct 2005 | Tse Lee | Modify the PnrGen return message Element from "Reply" to "Response" |
| | 21 Oct 2005 | Tse Lee | Add Market Matrix table for the reference load. Add the CCITIN response error. |
| | 24 Oct 2005 | Tse Lee | Add the Data Editor for the reference loader |
| | 25 Oct 2005 | Tse Lee | Add the Internet Communication Protocol for the loading file convention |
| | 26 Oct 2005 | Tse Lee | Modify reference table key types fro the number format. |
| 1.3 | 28 Oct 2005 | Tse Lee | Change the format of the SID |
| | 29 Oct 2005 | Tse Lee | Modify the layout of System Out content |
| | 31 Oct 2005 | Tse Lee | Add the Confirmation/Rejection information |
| 1.4 | 1 Nov 2005 | Tse Lee | Create the version on the CCITIN and PNRGEN request |
| | 3 Nov 2005 | Tse Lee | Modify the IATA as optional field |
| 1.5 | 9 Nov 2005 | Tse Lee | Add the preferred date in the request direction as optional field |
| | 10 Nov 2005 | Tse Lee | Add the Office Code in the schema and example of the PnrGen request |
| | 30 Nov 2005 | Tse Lee | Correct the 3.5.2.2.2.5 parentCode, 3.5.2.2.6 response tag, and CCITIN response example.<br>Provide the descending nested order example for the classes display. PNRGen response data fields provide. |
| 1.6 | 12 Dec 2005 | Tse Lee | Provide the Segment portion of CCITIN for one carrier (3.5.2.2.8) Adding Parent class code and city point of sale.<br>Modify the primary key definition of the market fare table. |
| 1.7 | 16 Dec 2005 | Tse Lee | Add the two fields (MinSeatRqstd, Negoflag) on the Request load mainly for the negotiation module |
| | 16 Dec 2005 | Tse Lee | Update all the examples such that if there any optional field without any value, G6 will not send this attribute out. Also update all the data fields with explanations on all the optional fields. |
| | 19 DEC 2005 | Tse Lee | Correct the MinSeatRqstd to be optional. Negoflag values will be valid only from 0, 1, 2, 3, and 4. |
| 1.8 | 19 DEC 2005 | Tse Lee | Add the Office Code to the PnrLoad 3.3.2.2 |
| | 20 Dec 2005 | Tse Lee | Change the PNRGen response 3.5.3.2 errMsg from required to optional. Remove the character definition on ". |
| | 21 Dec 2005 | Tse Lee | Change the errMsg in CCITIN from required to optional. |

| | | | |
|---|---|---|---|
| 1.9 | 28 Dec 2005 | Tse Lee | Negotiation input 3.4.1.2<br>Add the additional message on the confirm and rejection response |
| | 10 Jan 2006 | Tse Lee | 3.4.1.1 Add Trip Duration to the directional information of the Negotiation module.<br>Remove the segment information from the negotiation module. That segment information was designed for segment control carriers for the pricing inquiry purpose. |
| | 20 Jan 2006 | Tse Lee | Change the Note type to Char04 as requested by Carlos. |
| 1.10 | 6 Feb 2006 | Tse Lee | Add one additional column Country POS in the reference City POS table |
| 1.11 | 7 Feb 2006 | Tse Lee | 3.3.2.3 Acceptance interface creation.<br>Change the OptionID and SubOptionID as integer representation although the type is still char04.<br>Change the Quoted Fare to be optional in the RequestLoad as requested by Carlos. |
| 1.12 | 10 Feb 2006 | Tse Lee | Change the Acceptance loader SID format.<br>Direction and option level information are separated.<br>Change the accepted fare as optional.<br>StaleFalg information is still under design.<br>Correct the entire RequesDate format to be YYYYMMDD in the Request load, Negotiation, and Acceptance. |
| | 14 Feb 2006 | Tse Lee | Change the Option ID to the number type on the Negotiation and Acceptance modules. |
| | 28 Feb 2006 | Tse Lee | Add the special note for the PnrLocator type based on the request from one carrier. That carrier has old G52 PNRLocator loading format. It contains unnecessary spaces. G6 will trim those spaces if necessary. |
| | 2 Mar 2006 | Tse Lee | 3.3.2.1 Make the City POS and District POS as required fields in the Ad Hoc Request Load. |
| 1.13 | 2 Mar 2006 | Tse Lee | Series Request SID draft creation on the Data fields |
| | 6 Mar 2006 | Tse Lee | Add the negotiation file naming and acceptance naming at the file description 4.2.3 |
| | 14 Mar 2006 | Tse Lee | Add the Kebut three fields in the Series request header |
| | 21 Mar 2006 | Tse Lee | Provide the schema and example for the Series module. Correct some spelling errors. |
| | 22 Mar 2006 | Tse Lee | Provide the schema for the Kebut Request |
| | 23 Mar 2006 | Tse Lee | Change 3.4.1.2. Change the three optional fields of Loc as Series only |
| | 24 Mar 2006 | Tse Lee | 3.5.3.1 Change the NoteType and Value attribute name of <PRO> to noteType and value. The CCITIN and PNRGen schema are generated by using the Relax NG tool. |
| | 27 Mar 2006 | Tse Lee | Change the Note from the Series Request level to the Pattern level. |
| | 28 Mar 2006 | Tse Lee | Add the description of the optional fields |
| | 28 Mar 2006 | Tse Lee | Correct the Series Segment comment |
| | 31 Mar 2006 | Tse Lee | Correct the "accepted delete" mark on the SID |
| | 11 Apr 2006 | Tse Lee | Provide the primary key constrain for the Series Loader |
| | 12 Apr 2006 | Tse Lee | Correct Series fare information SFI example error |
| 1.14 | 21 Apr 2006 | Tse Lee | Add the exchange rate on the PnrGen PRM record |
| | 1 May 2006 | Tse Lee | Remove red color reminder from the document |
| 1.15 | 2 May 2006 | Sanjay Datta | Add Seg AU in the Ccitin Reply Segment Class Section for CITP. |
| 1.16 | 2 May 2006 | Tse Lee | Change the NestedAU of the Segmentss of CCITIN response as optioal<br>Add the schema and example for the nestAU for CITP CCITIN response<br>Add the Exchange Rate table. |
| 1.17 | 15 May 2006 | Tse Lee | Change the fare (quoted fare,local fare, ang marginal fare) from NUM 08 to NUM 10.<br>Add the local fare in the PNR Loader SID interface. |
| | 17 May 2006 | Tse Lee | Correct the example of Trip Duration. It does not apply to direction 1. |
| | 17 May 2006 | Tse Lee | Correct the Adhoc XSD syntax errors. |
| | | | |

| | | | |
|---|---|---|---|
| | 17 may 2006 | Tse Lee | Change the NUM02 – NUM10 type definition from maxLength to totalDigits<br>Change NUM05 and NUM08 type from unsignedShort to UnsignedInt<br>Change NUM10 to unsignedLong |
| | 22 May 2006 | Tse Lee | Correct the Adhoc example MinSeatsRqst to MinSeatRqst<br>Chane the Adhoc Schema<br>Remove the empty element from the Adhoc example. |
| | 23 May 2006 | Tse Lee | Change the XSD PreferredFlightFlag and PnrGenFlag |
| | 24 May 2006 | Tse Lee | Change the PNRLocator and PrimPnrLocator to be optional in the RHP of Ad Hoc request loader. |
| | 30 May 2006 | Tse Lee | Correct the SDT missing DptNum in the example<br>Correct the misspelling on the PreferredFalg in the Series Loader example |
| | 31 May 2006 | Tse Lee | Add the Num01Type as the reference type<br>Correct CCITIN response example on the Flight Leg closing tag<br>Correct CCITIN response schema Option tag<br>Correct CCITIN response schema leg compartmrnt data type<br>Correct CCITIN response schema departureDate spelling |
| 1.18 | 1 June 2006 | Tse Lee | Origin, Destination, and Class on the Series Segment (SFP).<br>ProcessId is for the Kebut ID communication on SDT.<br>ProcessId added for the Confirmation |
| 1.19 | 1 June 2006 | Tse Lee | Add the optional attribute notAvialable in the flight element of CCITIN response.<br>Change the CCITIN response file attributes. Specificy the required attributes for the necessary fields. Correct CCITIN response schema error. |
| | 2 June 2006 | Tse Lee | PNRGen request example correction on the <Request> closing tag.<br>Add the use = "required to the PNRGen Respopse schema |
| | 6 June 2006 | Tse Lee | Fix the exchange rate example error |
| | 7 June 2006 | Tse Lee | Fix the Series Loader example group type S |
| | 9 June 2006 | Tse Lee | Change the CCITIN Reply schema<br>Change the Air Craft String to be option under the CCITIN Response Leg |
| | 12 June 2006 | Tse Lee | Use the Relax NG to generate the PNRGen response file |
| 1.20 | 27 June 2006 | Tse Lee | Add the nonControlFlag to the CCITIN flight response<br>Modify the ccitin response xsd |
| | 05 July 2006 | Tse Lee | Add the comment on the Series Loader Seggment. The Class Code is reserved for the kebut procxess. |
| | 28 June 2006 | Tse Lee | Correct the description of the PNR local fare |
| 1.21 | 2 Aug 2006 | Tse Lee | 3.4.1.1<br>HFO record add FlightInfo and ConnectionCities<br>Add HFF record to the negotiation |
| 1.22 | 6 Aug 2006 | Tse Lee | 3.3.2.4<br>1. Move the GroupShell ID and PNRLocator from the header level.<br>2. Add the GroupShell ID and PNRLocator on the departure level |
| | 10 Aug 2006 | Tse Lee | 3.4.1.1 Change the Fligh IN comment and change the connection cities attribute as optional |
| 1.23 | 10 Aug 2006 | Tse Lee | 3.4.1.2 Add the Group Shell ID on the confirmation file for Kebut request<br>3.3.2.4 remove the GroupShell ID from the series departure level. |
| 1.24 | 21 Aug 2006 | Tse Lee | 3.3..2.4 Add the finished flag for the Kebut request termination indicator.<br>3.3.2.4 Remove GroupShellID and PnrLocator from the XSD |
| 1.25 | 8 Sep 2006 | Tse Lee | Reference table Peak/Off-Peak design |
| 1.26 | 21 Sep 2006 | Tse Lee | Remove the Finished from the Series Header. |
| | 25 Sep 2006 | Tse Lee | Provide the schema for the Peak/Off Peak reference loader |

| | | | |
|---|---|---|---|
| | 2 Oct 2006 | Tse Lee | 3.5.3.1 change the PNRGEN district POS as optional |
| | 2 Oct 2006 | Tse Lee | 3.5.3.1 modify the schema for district POS |
| | 4 Oct 2006 | Tse Lee | 3.5.3.1 modify the schema and description of cityPOS as optional |
| | 6 Oct 2006 | Tse Lee | Change the pLoadPeak to pLoadPOP and file extention file to POP |
| | 13 Oct 2006 | Tse Lee | 3.3.2.1 Add the comment for the PNR request on the preferred flight information |
| | 17 Oct 2006 | Tse Lee | 3.3.2.1 Add the comment for the RFS. It is mandatory for the PNR Request. |
| | 04 Dec 2006 | Tse Lee | Relax the restrictions of unique Note Type. The loading note can have multiple note type. |
| | 08 Jan 2007 | Tse Lee | Fix the xsd schema for the Acceptance Load. The note type is unbounded.<br>Also, fix the Negotiation xsd for the Note type and other attributes |
| | 10 Jan 2007 | Tse Lee | Add the Char04, Char11, Char26 type and correct the Negotiation example |
| | 11 Jan 2007 | Tse Lee | Change the Confirmation schema and example |
| 1.27.1 | 11 Jan 2007 | Tse Lee | Introduce the airlineVenderID in the CCITIN and PNEGEN request<br>Change the schema of CCITIN and PNRGen |
| | 19 Jan 2007 | Tse Lee | Add the CCITIN error message in the schema and change the PNRGen error message size to 50 |
| | 31 Jan 2007 | Tse Lee | Provide the reference to the web service specification |
| | 5 Feb 2007 | Tse Lee | Remove the carrier specific referenc in the change history. |
| | 7 Mar 2007 | Tse Lee | Correct the number type NUM02 inconsistency in the request file |
| | 8 Mar 2007 | Tse Lee | Add the comments on the leading spaces of the loading data |
| | 9 Mar2007 | Tse Lee | Add the NUM04 type to the basic type. Correct the basiic type definitions. |
| | 12 Mar 2007 | Tse Lee | Change the market fare table search type to Flag01 for consistency.<br>Change special fare table comaprtment code to Char 02<br>Change the flag type to flag 01 type on the schema for most of the flags on the request header. Change the request type flag from flag 01 to flag<br>Add Char256 type for the note text<br>Add Char05Type for the country POS type<br>Change the PNRLoad market destination data type to CityType<br>Change cancellation flag and outbound flag to flag 01 for the pneload.<br>Change the note type size to 256 for all the notes<br>Change the flge type to flag 01 type for Series Header data (partial confim, preferred flag…)<br>Change the negotiation on the option selected flag to Num01<br>Rewrite the Series Loader schema. |
| | 13 Mar 2007 | Tse Lee | Add additional comments on the flight number regarding the 5 digits. |
| | 14 Mar 2007 | Tse Lee | Remove the city type minimum 3 characeters restriction.<br>Change district type to country type |
| | 22 Mar 2007 | Tse Lee | Remove the additional comment from the range number on Ad Hoc Request<br>Change the |
| 1.27.2 | 22 Mar 2007 | Tse Lee | Add two additional attributes on the 3.5.3.1 PNRGen for the GHI feature. |
| | 27 Mar 2007 | Tse Lee | Remove the control characters from the SID.<br>Correct the confirmation note values to value<br>Correct the XSD of confimation and rejection<br>Remove maxOccurs="1" for the optional element<br>Remove the commment " or pass a balnk value…"<br>Remove Time06<br>Use the new XSD for the PNRLoad.<br>Correct the PNRLoad example |

| | | | |
|---|---|---|---|
| | 27 Mar 2007 | Tse Lee | Remove the schemalocation from the XSD |
| | 28 Mar 2007 | Tse Lee | Add the recordset description for Ahdoc., Acceptance, Series, PNRGen, and PNR loader. |
| | 29 Mar 2007 | Tse Lee | CCITIN and PNRGen request XSD type addition.<br>Correct the orientation to Symbol02Type even G6 sends out only one character<br>Change the NUM XSD to handle the +/- sign of values. |
| | 2 April 2007 | Tse Lee | Correct the Symbol description maximum from 1 to 2. |
| | 19 April 2007 | Tse Lee | Add the comment for the CCITIN day offset response. |
| | 24 April 2007 | Tse Lee | Add the comment on the nestedAU of the CCITIN segment class. |
| | 26 April 2007 | Tse Lee | Add the note for the Time Limits example on the PNRGen note |
| | 27 April 2007 | Tse Lee | Change the PreferredFlightFlag to PreferredFlag on the SHG schema of the Series loader |
| | 3 May 2007 | Tse Lee | Remove the transactional data schemas from SID.<br>Rename the time limits to name firming. |
| | 4 May 2007 | Tse Lee | Change the name firming PRO type t o NFM |
| | 23 May 2007 | Tse Lee | Change the comment of the accepted fare in the negotiation. |
| | 7 June 2007 | Tse Lee | Remove the reference schema from the SID |
| | 12 June 2007 | Tse Lee | Change the Program Name from required to optional |
| | 1 Aug 2007 | Tse Lee | Add the comment to the PMG and constraint to match the XSD as required |
| | 2 Aug 2007 | Jay Goldberg | Modified Agency table. Added Traffic Area and Region tables and dataloader information. Added Country POS dataloader. |
| | 6 Aug 2007 | Tse Lee | Mofify the AU and Segment limit constraint on the Segment Class of CCITIn response |
| | 12 Sep 2007 | Tse Lee | Change the quoted fare as optional in the Series Loader design |
| | 13 Sep 2007 | Tse Lee | Add the comment for the CustType in the PNRLoad for the non-existing value. Pass 0.<br>Add the comment for the acceptance not regarding the duplicate nots. |
| | 28 Nov 2007 | Tse Lee | Change the title Internet Communication Protocol to the Interface Communication protocol. |
| | 29 Nov 2007 | Tse Lee | Add the comment on the PNR rrejection system. The group shell id will be 0 in this case. |
| 1.27.3 | 12 Dec 2007 | Tse Lee | Add the realFare and localFare for the PNR level on the PNRGen request 3.5.3. |
| 1.27.4 | 31 Jan 2008 | Tse Lee | Add the deal code and term tables<br>Add the AOL fare reference<br>Add the deal indicator and trip type on the Ad Hoc header<br>Add the stop over indicator and stop over cost on the Ad Hoc direction |
| | 8 Feb 2008 | Tse Lee | Change OAL pos definition as class POS and change the size to char 10 |
| 1.27.5 | 8 Feb 2008 | Tse Lee | Add the Deal message to the Series Loader on the header and direction level |
| | 13 Feb 2008 | Tse Lee | Change the Stop over cost to NUM10 on the Terms (confirmation/rejection) |
| | 13 Feb 2008 | Tse Lee | Change the Term data field from locator to id. |
| | 14 Feb 2008 | Tse Lee | Change the Deal type to identity and add back the deal type |
| 1.27.6 | 20 Feb 2008 | Tse Lee | Change the OAL table description according to a new design |
| | 22 Feb 2008 | Tse Lee | Change the example for the OAL fare. Change the carrier, compartment, class type definitions. |
| | 25 Feb 2008 | Tse Lee | Change the Deal Indicator as Char 02 in the request header and series header |
| | 29 Feb 2008 | Tse Lee | Change the terms on the reply file. Put capital character on the first letter of element name. |
| 1.28.0 | 11 Mar 2008 | Tse Lee | Series loader changes (SetNum and SeasonIndicator) for the ORW module. |
| | 14 Mar 2008 | Tse Lee | Change the start date as required in the OAL table |

| | | | |
|---|---|---|---|
| | 17 Mar 2008 | Tse Lee | Provide the comment on the optional End Date on the OAL table. |
| | 25 Mar 2008 | Tse Lee | Force the OAL table to have the carrier code values. |
| | 26 Mar 2008 | Tse Lee | Correct the start date and end date definition for travel departure date range |
| | 28 Mar 2008 | Tse Lee | Correct PRM description. It is one per group shell |
| | 8 Apri 2008 | Tse Lee | Add the exchange rate formula on the exchange rate table |
| | 13 May 2008 | Tse Lee | Add the comment on the trip duration |
| | 20 May 2008 | Tse Lee | Add the comment to the CCITIN noControlFlag response file |
| 1.29.0 | 30 May2008 | Tse Lee | Add additional field percent_name as optional on the gx_country_pos |
| | 10 June 2008 | Tse Lee | Modify the PNRLoad input file and make the group type as optional |
| | 30 June 2008 | Tse Lee | Add one additional field Preferred Option for the negotiation file |
| | 03 July 2008 | Jay Goldberg | Add CRS to request header and pnrgen request. Add reject_flag to pnrgen response. |
| | 17 July 2008 | Tse Lee | Expand the User iD to character length 128 in the PNRGen request file |
| 1.30.0 | 08 Aug 2008 | Tse Lee | Add the Program name to the Negotiation file |
| | 25 Aug 2008 | Tse Lee | Add the comment on the PNRLOad day offset value to include O&D |
| | 27 Aug 2008 | Jay Goldberg | Add AgentURL to AdHoc and Series Loaders |
| | 30 Nov 2008 | Tse Lee | Remove the transaction description on 4.2.2 |
| | 15 Dec 2008 | Tse Lee | Correct the 3.3.1.11.2 comment |
| | 22 Jan 2009 | Tse Lee | Correct PMG comment for the Group Type of PNRLoad |
| 1.31.0 | 11 Feb 2009 | Tse Lee | Add Segment NestedAvailability fields in the class level<br>Add allowAvailability in the segment compartment level |
| | 26 Feb 2009 | Tse Lee | Add another type for the Open Jaw for the trip type. |
| | 5 Mar 2009 | Tse Lee | Correct the comment update date |
| | 2 Apr 2009 | Tse Lee | Update the PNRLoad LCU and Local fare information |
| | 21 Apri 2009 | Tse Lee | Correct CCITIN request Flight element (destination and date).<br>Correct CCITIN response Option to optional if there is any error.<br>CCITIN response segment compartment and class elements are required. |
| 1.32.0 | 12 May 2009 | Tse Lee | Add market martrix date range table<br>Add market fare adjustment table |
| | 29 June 2009 | Tse Lee | Add 3.5 for the Reservation Migration content |
| | 16 July 2009 | Jay Goldberg | Enhancing OAL Fare loader |
| 1.33.0 | 21 July 2009 | Tse Lee | Add the oalAvailability at the leg compartment |
| | 23 July 2009 | Cecil Ureña | New columns in the Market Fare table. |
| | 23 July 2009 | Tse Lee | Atddditional comment on the Market fare<br>DepartStartTime, DepartEndTime, Frequency, PaxHi,and are added PaxLo<br>Change the frequency type description.<br>Add the comment for PaxLo and PaxHi. |
| 1.34.0 | 04 August 2009 | Javid Jamae<br>Jay Goldberg | Fixed typos<br>Moved the Deal element inside of the PNR element for PNRGen Request and made it support arbitrary deal items defined by keys/value pairs.<br>Moved DealIndicatorFlag and TripType from the Series Request Load SHG record to the SPG record (from series to pattern level).<br>Deprecated the confirmation/rejection message and created a new confirmation/rejection message structure that supports terms and conditions at the pattern level for series rather than supporting a single terms and conditions structure for an entire series. Confirmation/rejection messages now support arbitrary terms and conditions items defined by key/value pairs. |
| 1.35.0 | 13 August 2009 | Cecil Ureña | Created section for Deposit Deadline loader.<br>Added dataloader file examples. |

| | | | |
|---|---|---|---|
| 1.36.0 | 18 August 2009 | Hector Villarreal | Created section for Name firming Deadline loader.<br>Added dataloader file examples. |
| | 9 September 2009 | Cecil Ureña | Modified deposit deadline section<br>Fixed wording of Booking From and Booking To columns. |
| | 10 September 2009 | Tse Lee | Remove the SID version back to 1.36.0 since there is no schema change. |
| | 5 October 2009 | Tse Lee | Remove the rule process reference for the PNR Loader and Ad Hoc Request Loader post proceee.<br>Modify the PMG note on the PNR Loader for any Group Shell created outside G6 |
| | 12 October 2009 | Tse Lee | Change the Exchange Rate loader from Insert to Insert/Update in the document 3.3.1.12 |
| 1.37.0 | 13 October 2009 | Tse Lee | Add the Local fare on the negotiation 3.4.1.1 |
| 1.38.0 | 20 October 2009 | Tse Lee | Change Frequency field from optional to required 3.3.1.7 Market Fare table<br>Change Frequency field from optional to required 3.3.1.17 OAL Fare table<br>Change TripOrign field from optional to required 3.3.1.17 OAL Fare table<br>Change TripDestination field from optional to required 3.3.1.17 OAL Fare table<br>Change StartTime field from optional to required 3.3.1.17 OAL Fare table<br>Change PaxMin field from optional to required 3.3.1.17 OAL Fare table |
| | 27 October 2009 | Tse Lee | Change comment on the 1.38.0<br>Change OAL example |
| | 20 January 2010 | Tse Lee | Add comments on the request header for ApExclusive field in section 3.3.2.1.2.1<br>Add the default value for this field. |
| | 20 January 2010 | Tse Lee | Change the OAlL fare task name and file extention |
| | 8 March 2010 | Tse Lee | Additional comment on the loading date of PNR Loader 3.3.2 |
| | 15 March 2010 | Tse Lee | Additional comment on the PrimaryPNRLocator of Ad Hoc Rerquest Loader 3.3.2.1 |
| 1.39.0 | 5 April 2010 | Jay Goldberg | Added Ancillary Revenue and Via for Market Fare |
| | 16 April 2010 | Tse Lee | Modify the Ancillary Revenue name from CHAR size 256 to 24 under 3.3.1.7.<br>Change the Via from Char 05 to City<br>Change the Order of Via |
| 1.40.0 | 19 April 2010 | Tse Lee | Add the Ancillary Revenue on PNR generation interface 3.5.3.1 |
| 1.41.0 | 17 May 2010 | Tse Lee | Ancillaryrevenue element added in 3.3.2.2.2.2 for PNR data loader |
| | 15 November 2010 | Tse Lee | MinSeatRqstd comment provided shoule be greater than 0 on section 3.3.2.1.2.1 |
| | 11Jan2011 | Jay Goldberg | Change deal value length to 1024. |
| 1.42.0 | 12 Jan 2011 | Tse Lee | Schema changes on the PNR Genration request from size 256 to 1024 on section 3.5.3.1.2.6. |
| | 25 Jan 2011 | Tse Lee | 3.3.1.11 AdjPercentage comment correction |
| 1.43.0 | 16 March 2011 | Tse Lee | 3.4.1.1.2.4 ClsCode and ClassPOS added on the negotiation module |
| 1.44.0 | 16 March 2011 | Tse Lee | 3.3.2.3.2.3.1 Add a new block AFF for the acceptance module |
| | 21 March 2011 | Tse Lee | Correct the HFF description from HHF to HFF |
| | 21 March 2011 | Tse Lee | Update HFF example on the new ClassCode field |
| 1.45.0 | 30 March 2011 | Tse Lee | 3.3.2.3.2.3.1 Modify the AFF with the latest design |
| 1.46.0 | 6 April 2011 | Tse Lee | Update negotiation flightInfo size from char 26 to char 128 |
| 1.47.0 | 29 April 2011 | Tse Lee | 3.5.2.2.2.2 Add one additional arrivalDate on the CCITIN flight response. |
| 1.48.0 | 2 May 2011 | Tse Lee | 3.3.2.2.2.2 Add the outstation to the PMG |
| 1.49.0 | 17 August 2011 | Tse Lee | 3.3.2.4.2.5 Relax the <SFP> as a required constraint |
| 1.50.0 | 2 September 2011 | Tse Lee | 3.5.4 PNR Refresh interface |
| 1.50.0 | 19 September 2011 | Jay Goldberg | Add Discount Percent to City POS loader |
| 1.51.0 | 20 September 2011 | Tse Lee | Correct the version number sicne Schema has changed |

| | | | |
|---|---|---|---|
| 1.52.0 | 26 September 2011 | Jay Goldberg | Add Discount Percent to AdHoc and Series loaders |
| 1.53.0 | 3 October 2011 | Tse Lee | 3.5.3.1.2.1 Add the adjustedRealFare on the PNR Level. |
| 1.54.0 | 3 October 2011 | Tse Lee | 3.5.2.2.2.4 Add the groupBooked on the CCITIN response Leg Compartment |
| 1.55.0 | 19 Octoboer 2011 | Jay Golderg | Remove Discount Percent from City POS Loader and add a City POS Discount Loader |
| 1.56.0 | 28 November 2011 | Sonny Tran | Add an optional Local Fare to Acceptance loader and require/optional clarifications to the comments.<br>The new local fare is added to the Option recordSet: HFO |
| 1.57.0 | 16 January 2012 | Tse Lee | 3.4.1.1.2.3 Add `ArrDayOffset` |
| | 18 January 2012 | Tse Lee | Move the `ArrDayOffset position` |
| | 1 Febuary 2012 | Tse Lee | Make the classCode optional on the negotiation |
| 1.58.0 | 25 April 2012 | Tse Lee | Add the Markerting Flight Fare loading table |
| | 28 June 2012 | Tse Lee | Remove the CityPOs and DisPos comment for getting value from Agency table for the Ad Hoc and Series Request |
| 1.59.0 | 10 September 2012 | Tse Lee | Add a new flight capacity control data loader |
| | 11 September 2012 | Tse Lee | Add the reference data tag for flight group capacity control |
| | 17 September 2012 | Tse Lee | Put the key information to the flight group capacity table for the compartment |
| | 20 September 2012 | Tse Lee | Change some should wording |
| 1.60.0 | 23 October 2012 | Tse Lee | Add RestOfWorld on the Market Fare Origin table |
| | 24 Oct 2012 | Tse Lee | Provide comment for Rest of World flag |
| | 29 October 2012 | Tse Lee | Modify the Rest of World flag indicator comment |
| 1.61.0 | 19 November 2012 | Sonny Tran | Add keyAccount field to PNRGen Request in RecordSet Group |
| 1.62.0 | 17 April 2013 | Tse Lee | Add a new Reject Request loader |
| | 23 April 2013 | Tse Lee | Enhance the reject loader general description |
| | 24 April 2013 | Tse Lee | Minor change on the reject loader description |
| | 23 May 2013 | Tse Lee | Comment on the 10-digit fare |
| 1.63.0 | 11 September 2013 | Jay Goldberg | Added optional Compartment Code to HFO holding file record set |
| 1.64.0 | 24 November 2013 | Brent Franklin | Added adjustedCapacity to CCITIN Leg Compartment<br>Added totalAU to CCITIN Leg Compartment<br>Added Bid Price to CCITIN Leg Compartment<br>Added Aircraft Configuration List to CCITIN Leg<br>Added Inventory Map to CCITIN Leg |
| 1.65.0 | 27 November 2013 | Brent Franklin | Added odAvailability to CCITIN Segment Class |
| 1.66.0 | 20 December 2013 | Brent Franklin | Added Bid Price index in CCITIN Leg Compartment |
| 1.67.0 | 24 January 2014 | Brent Franklin | Added EFT to CCITIN Option<br>Added EFT to CCITIN Flight<br>Added Marketing Carrier Code and Marketing Flight number to CCITIN Flight<br>Added Aircraft Type to CCITIN Flight |
| 1.67.1 | 29 January 2014 | Naresh Pakalapati | Added optional fields Carrier Code and Flight Number to MFG market fare data |
| 1.68.0 | 05 February 2014 | Brent Franklin | Added Country POS to CCITIN Request |
| 1.69.0 | 25 March 2014 | Sonny Tran | Added optional field SetNum to RFS record in Request Loader and setNumber in Ccitin request under prefer flight data |
| 1.69.0 | 25 March 2014 | Sonny Tran | Added rqstID to CCITIN Request |
| 1.69.0 | 25 March 2014 | Sonny Tran | Added fcstDmdLF to CCITIN Leg Compartment |
| 1.70.0 | 15 July 2014 | Naresh Pakalapati | Change the terms on the series reply file. Moved terms and conditions from pattern level to locator level. |

| | | | |
|---|---|---|---|
| 1.71.0 | 6 Aug 2014 | Jay Gooldberg | Added fields to the CCITIN request direction: direction_informative, origin_informative, destination_informative, compartment, via, OALonly |
| 1.72.0 | 9 Aug 2014 | Mac McEacharn | Added allowOAL and directFlightsOnly to the request direction |
| 1.73.0 | 11 Aug 2014 | Brent Franklin | Added groupCapacityPercent to CCITIN Leg Compartment response |
| 1.74.0 | 08 Feb 2018 | Chenna Maduru | Added IMPLEMENT_DATE and DEACTIVATE_DATE to Speicia fare |

# Table of Contents

# 1. Overview

## *Scope*

The PROS Group 6 revenue Management System is an airline revenue management system which allows airlines to control and perform an economic evaluation of a group passenger request. This document, the System Interface Document (System SID) describes the various inputs used to transact with the PROS Group 6 Revenue Management System. This data is required to support the evaluation, confirmation and post acceptance maintenance. Note that this document identifies tables and data required at the time of installation. Other tables and data files may be required during various phases of this project and they will be identified by the SRD and SID of those components as and when they are required.

## *Purpose*

This initial draft is a part of the initial discussion regarding the use of the Web Services technology for the communication between G6 Web Based Application and RES interface.

Please note that this is only a draft and some of the items may change as we go into detail requirements.

The purpose of this document is to provide an idea on the type of data that would be passed between RES and the PROS Group 6 revenue Management System using the Web Services with the standard HTTP protocol. The data exchange format will follow the XML schema format that is described in this document.

The PROS Group 6 SID provides:

- General Format Requirements
- The Purpose of Each Record (where necessary)
- The Required Order Within Each Group of Related Records
- The Definition of Each Data Element (where necessary)

This document serves as the primary reference for the interfaces to PROS Group 6 Revenue Management System. In addition, the PROS Revenue Management staff will be available to work closely with the project teams during various stages of implementation. Some of the support functions performed by PROS staff on site include:

- Assisting clients in installing codes and any patches or upgrades as they arise
- Configure environment, and setting up production and test environments
- Setup System Parameter Tables
- Data Loading and Testing
- Data Analysis
- Monitor Loading Process

The Airline is responsible for providing the required data from their own data sources or from a third party. This includes building any data extraction programs needed to extract from the Airline or from third party sources and formatting the data in the exact manner required by the PROS Group 6 Revenue Management System. The Airline is also responsible for validating, accuracy and completeness of the data prior to transmitting the data to the PROS Group 6 System.

## *System Context*

# 2. Interface Load Formats

### *Interface Load Formats*

- All data will be composed of ASCII characters.
- The following are valid formats for nullable fields:
    - <TAG></TAG> (no spaces between the start and end tag)
    - <TAG/>
    - Do not include the tag at all if it is an optional field
- Blank lines will be ignored.
- All file formats must comply the rules and standards of the Extensible Markup Language (XML) 1.0 set forth by document REC-xml-19980210. This document specifies a syntax created by subsetting an existing, widely used international text-processing standard (Standard Generalized Markup Language, ISO 8879:1986I as amended and corrected) for use on the World Wide Web. Please see http://www.w3c.org.
- All files must begin with the following Root Header:
- <?xml version="1.0" ?>
- Every file consists of one Dataset, containing one or more Record sets with identical schema structure.
- Each dataset is constructed of a Dataset Header, Record sets and Dataset Footer
- Each Record set is constructed of a Record set Header, Data Fields and Record set Footer.
- Each dataset must have a pair that consists of a Dataset Header and a Dataset Footer. The Dataset Header must be the first record of the dataset. The Dataset Footer must be identical to the Header with a slash (/) between the leading bracket and the name, as shown:
- <DatasetName> - Dataset Header
- </DatasetName> - Dataset Footer
- The Dataset Footer must be the last record in the dataset.
- Each record set must have a Record set Header and Record set Footer. The Record set Header must be the first record of the record set. The Record set Footer must be identical to the Header with a slash (/) between the leading bracket and the name, as shown:
- <RecordsetName> - Record set Header
- </RecordsetName> - Record set Footer
- The Record set Footer must be the last record in the record set.
- Each field must be presented on a new record and must have a Start-Tag before and an End-Tag after the value. The End-Tag must contain a name that echoes the field's name as given in the Start-Tag, preceded by a slash (/), e.g. <FieldName>Value</ FieldName>
- No quotation should be used for the values.
- Trailing spaces will be truncated during the load.
- Leading spaces will not be considered a valid input and must be avoid. E.g. <FieldName> Value</ FieldName> must be avoided. The correct input must be <FieldName>Value</ FieldName>.

Example 1:

```
<?xml version="1.0" ?>
<DatasetName>
<RecordsetName>
<FieldName1>FieldValueA11</FieldName1>
     <FieldName2>FieldValueA12</FieldName2>
```

```
            …
</RecordsetName>
<RecordsetName>
<FieldName1>FieldValueA21</FieldName1>
     <FieldName2>FieldValueA22</FieldName2>

            …
</RecordsetName>
</DatasetName>
```

### *Processing/Data Rules*

### 2.1.1 Required

This indicates whether a specific field is required in the input records (Y = Yes, N = No). An empty field that is required will result in an error message in the log file and the record will be rejected. An empty field that is not required will result in a Null value in the database record.

Providing only spaces in a required field will result in an error during the load. The spaces will be truncated, as described above, resulting in a null field value.

### 2.1.2 Referential

A field that is validated using referential data requires a corresponding record to be inserted in the lookup table prior to inserting the record. For example, to insert a record in the Terminal table there must first be a record in the Region table that can be referenced.

A record will be rejected without corresponding referential fields and an error will be written to the log file.

### 2.1.3 Validity Check

A field with a data type validity check will be checked against that data type. A field with a range will be checked to fall within that range. If either of these checks fails, when required, the record will be rejected without corresponding referential fields and an error will be written to the log file.

#### *2.1.3.1 Pre-defined*

Values from a pre-defined list: these will be listed within the applicable section.

#### *2.1.3.2 Letters*

A-Z and a-z only

#### *2.1.3.3 Numbers*

0-9 only

#### *2.1.3.4 Alpha*

A-Z, a-z, 0-9, and the space character

#### *2.1.3.5 ASCII*

Freeform text field allowing all ASCII values, except for the following reserved characters:

ASCII code 34 – “
ASCII code 35 - #
ASCII code 37 - %
ASCII code 38 - &

ASCII code 39 – ‘
ASCII code 43 - +
ASCII code 60 - <
ASCII code 61 - =
ASCII code 62 - >

### *2.1.3.6 Decimal [a, b]*

a = number of places to left of decimal
b = number of places to right of decimal

### *2.1.3.7 Character (CHAR)*

The CHAR fields may include any ASCII printable character except double quote (") (ASCII 34).

### *2.1.3.8 Number (NUM)*

The *NUM* fields may only use the following characters: ‘0’ through ‘9’, ‘+’ and ‘-’ (ASCII 48 through 57,43 and 45). The ‘+’ character is redundant since all numeric values are assumed to be positive unless prefixed with the ‘-’ character. Leading zeros are not required.

Note: NUM 01 means the valid number is in the range 0-9.

NUM 02 means the valid number is in the range 0-99.

NUM 03 means the valid number to be in the range 0 – 999.

NUM 10 means the valid number to be in the range 0 – 2147483647.

For the NUM 02 or NUM 03, G6 expects the actual numeric number value. For example, 01 will not be valid for NUM 02. The actual number must be 1 instead. The NUM definition is for the reference. However, the Group System can take the number more than the defined range.

### *2.1.3.9 Date (DATE)*

The *DATE* field consists of eight characters between ‘0’ and ‘9’ (ASCII 48 through 57). The format is YYYYMMDD, where YYYY is the year, MM is the month (01 for January), and DD is the date.

Note: The DATE format is 8 digits to be year 2000 compliant.

### *2.1.3.10 Time (Time04)*

A *TIME* field consists of four numeric characters between ‘0’ and ‘9’ (ASCII 48 through 57) that will be interpreted as hours and minutes, HHMM. The valid range is 0000(midnight) to 2359 (one minute before midnight). Leading zeros need not be provided.

### *2.1.3.11 Flag (FLAG)*

A *FLAG* field consists of ‘0’ (ASCII 48) or ‘1’ (ASCII 49) in each position and are interpreted as:

- 0 Interpreted as OFF, NO, FALSE
- 1 Interpreted as ON, YES, TRUE

### *2.1.3.12 Symbol*

The *SYMBOL* field type is the same as CHAR field type with a maximum length of two (2) characters. It is used for identifying Airline’s booking class and aircraft compartments.

### *2.1.3.13 Carrier*

The *CARRIER* field is the same as the CHAR field type with a maximum length of three (3) characters. This field contains carrier codes.

### *2.1.3.14 Flight*

The *FLIGHT* field type is the same as the CHAR field type with a maximum length of five (5) characters. This field contains flight numbers. This field may contain non-numeric characters. G6 will send flight number with 5 digits when communicating to the outside world (O&D, CCITIN, PNRGen….). I.e. flight 1, flight 01, flight 001, and flight 0001 will all be converted to flight 00001. PROS encourages all the interfaces with flight number coming to G6 be the in the form of 5 digits. G6 will pad with zeros to the flight number and make it 5 digits if necessary. The interface (CCITIN request, PNRGen request, Negotiation…) receiving the flight number from G6 may need to be aware the flight format to avoid any potential conflict.

### *2.1.3.15 City*

The *CITY* field type is the same as the CHAR field type with a maximum length of five (5) characters. This field contains City or Airport codes.

### *2.1.3.16 Country*

The *Country* is same the as CHAR field type with a minimum length of two (2) characters and a maximum length of three (3) character.

### *2.1.3.17 Frequency*

The *FREQUENCY* field is the same as a series of FLAG fields with a maximum length of seven (7) FLAGs. *FREQUENCY* is used to identify the days of week over which an activity takes place (flight operation). This field must be coded as a series of seven (7) FLAGs with no intervening blanks. The first flag is for Monday, the last flag for Sunday. For example, if a flight operates Monday through Friday but not on Saturday and Sunday must be coded as 1111100. A flight operating on Saturday and Sunday must be coded as 0000011.

### *2.1.3.18 Locator*

*The Locator* field type is the same as the CHAR field type with a maximum length of twenty-one (21) characters. The PNR Locator in the Group System is different from the Reservation System's (RES) PNR locator, which usually has 6 characters. The PNR locator in the PROS Group System contains both the RES PNR locator **AND** creation date, and has the following format: "PNRLocYYYYMMDD". For example, if the PNR is created on May 30, 2000 and the RES' PNR locator is "AAAAAA", the PNR locator for PROS Group System is "AAAAAA20000530".

**IMPORTANT NOTE:** Some Airlines may require 2 Locators for designate a PNR: a RES locator and a GDS locator. The Group System will support this condition. The new format could have any one of the 3 options:

- "RESSTR/GDSSTRYYYYMMDD" or
- "RESSTRYYYYMMDD" or
- "/GDSSTRYYYYMMDD"

If a carrier doesn't use the GDS locator then use RESSTRYYYYMMDD (same as "PNRLocYYYYMMDD" format outlined in the description above).

PROS Group System will trim unnecessary spaces in the Locator input.

Example:

Case 1:

"H005T /Y9E9VI20060917" will have RES Locator "H005T" with five characters only.

The space before "/" will be removed in this example.

Case 2:

"HLQ4K /  20061220" will  have RES  Locator "HLQ4K". GDS Locator will not be generated.

"HLQ4K /  20061220" will have the same result as "HLQ4K20061220" when loading into PROS Group System.

#### *2.1.3.19 Defined Types*

The content of schema can be found at /config/interface/xsd/type.xsd file.

#### *2.1.3.20 Validity Checks during Batch Loading*

All validity checks indicated in this document are fully enforced when entering or modifying data through the PROFIT user interface. For performance reasons however, certain validity checks are necessarily relaxed during batch loads. The following rules apply during batch load processing:

- All general "alpha" or "letters" validity checks are replaced with an Oracle character type validity check, which allows any ASCII character. This rule does not apply in cases where only a specific letter range is allowed.
- All general "decimal" or "numbers" validity checks are replaced with an Oracle number validity check. This rule does not apply in cases where only a specific number range is allowed.

### 2.1.4 Processing Rules

Processing rules describe any modification to the data required upon loading.

### 2.1.5 Data Loader Behavior

A description of Loader configuration for updates, inserts, and delete/truncates is included here.  The following modes of operation are supported:

#### 2.1.5.1.1 Update, Insert

The Update/Insert mode of the data loader uses the idiom of first trying to update a row of data in the database; if the update fails (new record) then the row is inserted.  This method is useful for two reasons:

(a) Most of the data to be loaded is updates of the existing data, which will improve performance.
(b) The data is reference data for other tables, thus deleting a row would cause other data to be deleted.
(c) The existing data in the table is to be maintained.

#### 2.1.5.1.2 Insert, Update

The Insert/Update mode of the data loader uses the idiom of first trying to insert a row of data in the database; if the insert fails (existing record) then the row is updated.  This method is useful for two reasons:

(a) Most of the data to be loaded is inserts, which will improve performance.
(b) The data is reference data for other tables, thus deleting a row would cause other data to be deleted.
(c) The existing data in the table is to be maintained.

#### 2.1.5.1.3 Delete, Insert

The Delete/Insert mode of the data loader uses the idiom of first deleting the row of data, then inserting the row of data.  This mode can be used when:

(a) performance can be improved.
(b) no relationships exist that depend on the data.

#### 2.1.5.1.4 Truncate, Insert

The Truncate/Insert mode of the data loader uses the idiom of first truncating the entire table, then inserting the data from the file. The mode can be used when:

(a) performance can be improved
(b) no relationships exists that depend on the data
(c) the existing data can be deleted without consequences.

# 3. Interface Definitions

## *Terminology*

**Dataset** — A dataset contains a sequence of records. Rules are applied to the processing of records sequenced in a dataset.

**Record Header** — A record header is a keyword identifier that marks the beginning sequence of stream of subrecords and/or data fields.

**Root Header** — A root header is the first record of a file. It is the following tag:

```
<?xml version="1.0" ?>
```

**Dataset Header** — A dataset header is a tag, which is the first record in the dataset. It states the dataset name:

```
<DatasetName>
```

**Dataset Footer** — A dataset footer is the last record in the dataset. It is a tag with the dataset name, preceded by a slash:

```
</DatasetName>
```

**Record set Header** — A record set header is a tag, which is the first record in the record set. It states the record set name:

```
<RecordsetName>
```

**Record set Footer** — A record set footer is the last record in the record set. It is a tag with the record set name, preceded by a slash:

```
</RecordsetName>
```

**Sub-record** — A sub-record is a logical grouping of data fields separated by delimiters. A record may contain one or more sub-record types. A sub-record begins with a sub-record header followed by the data fields. Rules are applied to the processing of the data fields.

**Data Field** — A data field contains the raw input data, which may be characterized by data type and syntax. Rules are applied to the processing of the raw data fields.

**Start-Tag** — The beginning of every Data Field is marked by a Start-Tag:

```
<FieldName>
```

**End-Tag** — The end of every Data Field that begins with a start-tag must be marked by an End-Tag containing a single slash and a name that echoes the field name as given in the start-tag:

```
</FieldName>
```

**Field Value** — The text between the start-tag and the end-tag is called the field's value. That value will be loaded into the PROFIT database. The table name corresponds to the DatasetName and column name is the FieldName.

**Level *n* record** — Records may be "nested" more than two levels deep. In such cases, the record definitions will have a label indicating the hierarchy. The top level will be a "Level 1 record", and any

labeled "Level 2 record" will be subordinate (a subrecord) to the last preceding "Level 1 record" (etc., for "Level 3 record").

### *Special Requirements*

None

### *System Inputs*

### 3.1.1 Reference Data

The user will load the following datasets whenever any of the data has changed. Load date and time can be different for each data set. If a new data dependent set (say terminal) which depends on another data set (say supply point) is to be loaded, then the load order must respect the dependence.

| | Dataset Name | RecordSet Name | Record Description |
|---|---|---|---|
| 1 | CityPosData | PSC | City Point of Sale |
| 2 | DistPosData | PSD | District Point of Sale |
| 3 | MktOrgnDstn | MOD | Market Origin Destination |
| 4 | AgencyData | AGG | Agency information |
| 5 | CustomerData | NCT | Customer Type Information |
| 6 | MarketData | MOG | Market Origin Information |
| 7 | MarketFareData | MFG | Market Fare Information |
| 8 | SpecialFareData | MFS | Promotional Fare Information |
| 9 | MarketMatrixData | MMG | Weekend Definition in Market Information |
| 10 | ExchangeRateData | EXR | Exchange Rate |
| 11 | PeakData | MKT | Peak /Off-peak period data information |
| 12 | TrafficArea | TAD | Traffic Area Information |
| 13 | Region | RGD | Region Information |
| 14 | Country | CTD | Country POS Information |
| 15 | OALFareData | FVI | OAL fare information |
| 16 | DepositDeadlineData | DDL | Deposit deadline information |
| 17 | MktFltFareData | MFF | Marketing Flight Fare information |
| 18 | NameFirmingDeadlineData | NFD | Name Firming Deadline Data |
| 19 | CityPosDiscountData | PSD | City POS discount |
| 20 | FlightCapacityData | FCG | Flight Group Capacity Control |

### *3.1.1.1 City POS*

#### 3.1.1.1.1 Description

The city code for cities that are handled group requests.

#### 3.1.1.1.2 Data Fields

| | Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|---|
| 1 | CityPos | CITY | Y | | | Y | IAH | City Point of Sale |
| 2 | Description | CHAR 30 | N | | | N | HOUSTON | Name of the city |
| 3 | Country POS | CHAR 05 | N | | | N | US | Country Point of Sale. |

#### 3.1.1.1.3 Processing Rules

None.

#### 3.1.1.1.4 Loader Behavior

Insert/Update

#### 3.1.1.1.5 Schema

The content of schema can be found at /config/interface/xsd/citypos.xsd file.

#### 3.1.1.1.6 Example

```
<?xml version="1.0"?>
<CityPosData>
  <PSC>
   <CityPos>IAH</CityPos>
   <Description>HOUSTON</Description>
   </PSC>
   <PSC>
      <CityPos>LAX</CityPos>
      <Description>Los Angles</Description>
      <CountryPos>US</CountryPos>
  </PSC>
</CityPosData>
```

### *3.1.1.2 City POS Discount*

#### 3.1.1.2.1 Description

The extra discount that may be applied based on City POS and a date range.

#### 3.1.1.2.2 Data Fields

| | Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|---|
| 1 | CityPos | CITY | Y | | | Y | IAH | City Point of Sale |
| 2 | EffDate | DATE | Y | | | Y | 20060101 | This field contains the date starting from which the discount is valid<br>Format: YYYYMMDD |

| | Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|---|
| 3 | DisContDate | DATE | Y | | | N | 20060031 | This field contains the last departure date the discount is valid.<br>Format: YYYYMMDD |
| 4 | DiscountPercent | NUMBER 03 | Y | | | N | 15 | Discount percentage |

#### 3.1.1.2.1 Loader Behavior

Insert/Update

#### 3.1.1.2.2 Schema

The content of schema can be found at /config/interface/xsd/cityposdiscount.xsd file.

#### 3.1.1.2.3 Example

```
<?xml version="1.0"?>
<CityPosDiscountData>
  <PSD>
    <CityPos>IAH</CityPos>
      <EffDate>20060101</EffDate>
      <DisContDate>20060131</DisContDate>
      <DiscountPercent>10</DiscountPercent>
   </PSD>
   <PSD>
      <CityPos>LAX</CityPos>
      <EffDate>20060102</EffDate>
      <DisContDate>20060131</DisContDate>
      <DiscountPercent>15</DiscountPercent>
  </PSD>
</CityPosDiscountData>
```

### *3.1.1.1 District POS*

#### 3.1.1.1.1 Description

The district code table for countries that have group requests.

#### 3.1.1.1.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DistPos | CITY | Y | | Alpha | Y | IAH | District Point of Sale |
| Description | CHAR 30 | N | | ASCII | N | HOUSTON | Name of the district. Don't pass this field if there is no value. |

#### 3.1.1.1.3 Processing Rules

None

#### 3.1.1.1.4 Loader Behavior

Insert/Update

#### 3.1.1.1.5 Schema

The content of schema can be found at /config/interface/xsd/distpos.xsd file.

#### 3.1.1.1.6 Example

```
<?xml version="1.0"?>
<DistPosData>
  <PSD>
    <DistPos>IAH</DistPos>
    <Description>HOUSTON</Description>
  </PSD>
</DistPosData>
```

### *3.1.1.2 Market O&D*

#### 3.1.1.2.1 Description

The market origin and destination defaults to the first direction origin and destination of the group's travel itinerary. PROS Group System uses this information as tracking criteria for group behavior.

#### 3.1.1.2.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| MktODkey | NUM 05 | Y | | | Y | 1 | Market O&D key number determined by the client |
| Orgn | CITY | Y | | | N | JFK | Name of the origin city for this market |
| Dstn | CITY | Y | | | N | LHR | Name of the destination city for this market |
| Description | CHAR 30 | N | | | N | | Text. Don't pass this field if there is no value. |

#### 3.1.1.2.3 Processing Rules

None

#### 3.1.1.2.4 Loader Behavior

Insert/Update

#### 3.1.1.2.5 Schema

The content of schema can be found at /config/interface/xsd/mktod.xsd file.

#### 3.1.1.2.6 Example

```
<?xml version="1.0"?>
<MktOrgnDstn>
  <MOD>
    <MktODkey>1</MktODkey>
    <Orgn>JFK</Orgn>
    <Dstn>LHR</Dstn>
    <Description>test</Description>
  </MOD>
</MktOrgnDstn>
```

### *3.1.1.3 Agency Information*

#### 3.1.1.3.1 Description

Travel agency information table for agencies that have business with Airlines.

#### 3.1.1.3.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| IATA | CHAR 10 | N | | | N | A987654321 | IATA. It is not mandatory. However, either IATA or Office Code must exist. |
| Agency | CHAR 50 | Y | | | N | Work Travel Agency | Agency name |
| OfficeCode | CHAR 10 | N | | | N | HOU0034 | Office Code |
| City | CITY | Y | | | N | HOU | City Point of Sale |
| KeyAccount | CHAR 06 | N | | | N | | Key Account. The contact person information of this IATA. Don't pass this field if there is no value. |
| Department | CHAR 05 | N | | | N | EYG | Internal Department Identification |

#### 3.1.1.3.3 Processing Rules

None

#### 3.1.1.3.4 Loader Behavior

Insert/Update

#### 3.1.1.3.5 Schema

The content of schema can be found at /config/interface/xsd/agency.xsd file.

#### 3.1.1.3.6 Example

```
<?xml version="1.0"?>
<AgencyData>
  <AGG>
    <IATA>A987654321</IATA>
    <Agency>Work Travel Agency</Agency>
    <OfficeCode>HOU0034</OfficeCode>
    <City>HOU</City>
    <KeyAccount>test</KeyAccount>
    <Department>EYG</Department>
  </AGG>
</AgencyData>
```

### *3.1.1.4 Customer Type*

#### 3.1.1.4.1 Description

Customer type is the characteristic of the group. Airline identifies and standardizes the Customer type categories.

#### 3.1.1.4.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Key | NUM 10 | Y | | | Y | 1 | Customer key identifier |
| Description | CHAR 30 | Y | | | N | LUXURY | Customer description |

#### 3.1.1.4.3 Processing Rules

None

#### 3.1.1.4.4 Loader Behavior

Insert/Update

#### 3.1.1.4.5 Schema

The content of schema can be found at /config/interface/xsd/custtype.xsd file.

#### 3.1.1.4.6 Example

```
<?xml version=”1.0”?>
<CustomerData>
  <NCT>
   <Key>1</Key>
   <Description>LUXURY</Description>
   </NCT>
   <NCT>
      <Key>2</Key>
      <Description>SPORT</Description>
  </NCT>
</CustomerData>
```

### *3.1.1.5 Market Fare Origin*

#### 3.1.1.5.1 Description

Market fare reflects the ideal going rate ranges for each specified direction of travel in a particular market.

#### 3.1.1.5.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| City | CITY | Y | | | Y | JFK | City name of Market |
| MarketID | NUM 03 | Y | | | N | 1 | Market ID |
| RestOfWorld | FLAG 01 | N | | | N | 0 | Rest of world flag. It is designed to indicate this specific City and Market ID can use the Rest Of World market ID if the regular market fare retrieval is failed.<br>1: This City and MarketID can use the rest of world market ID if the regular market fare retrieval failed.<br>0: This City and MarketID can not use the rest of world marketID |

#### 3.1.1.5.3 Processing Rules

None

#### 3.1.1.5.4 Loader Behavior

Insert/Update

#### 3.1.1.5.5 Schema

The content of schema also can be found at /config/interface/xsd/mktorg.xsd file.

#### 3.1.1.5.6 Example

```
<?xml version=”1.0”?>
<MarketData>
  <MOG>

   <City>JFK</City>
   <MarketID>1</MarketID>
   <RestOfWorld>0</RestOfWorld>
```

```
</MOG>

<MOG>
   <City>DEL</City>
   <MarketID>2</MarketID>
   <RestOfWorld>1</RestOfWorld>
</MOG>
</MarketData>
```

### *3.1.1.6 Market Fare*

#### 3.1.1.6.1 Description

PROS Group System uses the market fare information as an additional source for group user during evaluation.

#### 3.1.1.6.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| MarketID | NUM 03 | Y | | | Y | 1 | Market ID |
| TravelOrgn | CITY | Y | | | Y | JFK | Travel Origin City |
| TravelDestn | CITY | Y | | | Y | LHR | Travel Destination City |
| SearchFlag | FLAG 01 | Y | | | Y | 0 | 0 = Travel Date Only<br>1 = Travel Date and Sales Date |
| EffectDate | DATE | Y | | | Y | 20000501 | Effective Date |
| DiscontDate | DATE | Y | | | N | 20001231 | Discontinue Date |
| ImplDate | DATE | Y | | | Y | 20010515 | Implementation Date |
| DeactDate | DATE | Y | | | N | 20010630 | Deactivation Date |
| Compartment | CHAR 02 | Y | | | N | Y | Compartment Code |
| MidWeekLowFare | NUM 10 | Y | | | N | 250 | Midweek Low Fare |
| MidWeekGoingFare | NUM 10 | Y | | | N | 350 | Midweek Going Fare |
| MidWeekAverageFare | NUM 10 | Y | | | N | 300 | Midweek Average Fare |
| WeekendLowFare | NUM 10 | Y | | Integer | N | 255 | Weekend Low Fare |
| WeekendGoingFare | NUM 10 | Y | | Integer | N | 355 | Weekend Going Fare |
| WeekendAverageFare | NUM 10 | Y | | Integer | N | 305 | Weekend Average Fare |
| DepartStartTime | CHAR 04 | N | | String | N | 0400 | Departure Start Time in 24 hr Format |
| DepartEndTime | CHAR 04 | N | | String | N | 0400 | Departure End Time in 24 hr Format |
| CrrCode | CHAR 03 | N | | String | N | ZZ | Carrier Code |
| FlightNum | CHAR 05 | N | | String | N | 00729 | Flight Number |
| Frequency | FLAG 07 | Y | | String | Y | 0000001 | 7-character frequency |
| PaxHi | NUM 04 | N | | Integer | N | 10 | High seat requested number |
| PaxLo | NUM 04 | N | | Integer | N | 10 | Low seat requested number |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Via | CITY | Y | | String | Y | FRA | Travel Via Point<br>Asterisk (*) means any via point. |
| <AR> | | N | | | | | Ancillary Revenues<br>Do not pass this field if the value is NULL. |

#### 3.1.1.6.3 Ancillary Revenue (AR)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Name | CHAR 24 | Y | | | Y | YQ | Name of Ancillary Revenue |
| Amount | NUM 10 | Y | | | N | 100 | Amount of Ancillary Revenue |

#### 3.1.1.6.4 Processing Rules

None

#### 3.1.1.6.5 Loader Behavior

Insert/Update

#### 3.1.1.6.6 Schema

The content of schema can be found at /config/interface/xsd/mktfare.xsd file.

#### 3.1.1.6.7 Example

```
<?xml version="1.0"?>
<MarketFareData>
  <MFG>
   <MarketID>1</MarketID>
   <TravelOrgn>JFK</TravelOrgn>
   <TravelDestn>LHR</TravelDestn>
      <SearchFlag>0</SearchFlag>
      <EffectDate>20000501</EffectDate>
      <DiscontDate>20001231</DiscontDate>
      <ImplDate>20010515</ImplDate>
      <DeactDate>20010630</DeactDate>
      <Compartment>Y</Compartment>
      <MidWeekLowFare>250</MidWeekLowFare>
      <MidWeekGoingFare>350</MidWeekGoingFare>
      <MidWeekAverageFare>300</MidWeekAverageFare>
      <WeekendLowFare>255</WeekendLowFare>
      <WeekendGoingFare>355</WeekendGoingFare>
      <WeekendAverageFare>305</WeekendAverageFare>
      <DepartStartTime>0300</DepartStartTime>
      <DepartEndTime>0600</DepartEndTime>
      <Frequency>0000001</Frequency>
      <CrrCode>ZZ</CrrCode>
      <FlightNum>729</FlightNum>
      <PaxHi>25</PaxHi>
      <PaxLo>15</PaxLo>
      <Via>*</Via>
      <AR>
        <Name>YQ</Name>
```

```
        <Amount>45</Amount>
      </AR>
      <AR>
        <Name>YR</Name>
        <Amount>458</Amount>
      </AR>
      <AR>
        <Name>XF</Name>
        <Amount>8</Amount>
      </AR>
  </MFG>
  <MFG>
      <MarketID>2</MarketID>
      <TravelOrgn>LHR</TravelOrgn>
      <TravelDestn>DEL</TravelDestn>
      <SearchFlag>0</SearchFlag>
      <EffectDate>20000501</EffectDate>
      <DiscontDate>20001231</DiscontDate>
      <ImplDate>20010515</ImplDate>
      <DeactDate>20010630</DeactDate>
      <Compartment>Y</Compartment>
      <MidWeekLowFare>250</MidWeekLowFare>
      <MidWeekGoingFare>350</MidWeekGoingFare>
      <MidWeekAverageFare>300</MidWeekAverageFare>
      <WeekendLowFare>255</WeekendLowFare>
      <WeekendGoingFare>355</WeekendGoingFare>
      <WeekendAverageFare>305</WeekendAverageFare>
      <DepartStartTime>0500</DepartStartTime>
      <DepartEndTime>0800</DepartEndTime>
      <Frequency>0000010</Frequency>
      <PaxHi>23</PaxHi>
      <PaxLo>18</PaxLo>
      <Via>*</Via>
      <AR>
        <Name>YQ</Name>
        <Amount>450</Amount>
      </AR>
      <AR>
        <Name>YR</Name>
        <Amount>4580</Amount>
      </AR>
    </MFG>
  </MarketFareFareData>
```

### *3.1.1.7 Special Fare*

#### 3.1.1.7.1 Description

A promotion fare is to define a specific market on a specific date range.

#### 3.1.1.7.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| MarketID | NUM 03 | Y | | | Y | 1 | Market ID |
| TravelOrgn | CITY | Y | | | N | JFK | Travel Origin City |
| TravelDestn | CITY | Y | | | N | LHR | Travel Destination City |
| EffectDate | DATE | Y | | | N | 20000501 | Effective Date |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| ImplDate | DATE | N | | | N | 20170501 | Implementation Date |
| DeactDate | DATE | N | | | N | 20170501 | Deactivation Date |
| DiscontDate | DATE | Y | | | N | 20171231 | Discontinue Date |
| Compartment | CHAR 02 | Y | | | N | Y | Compartment Code |
| SpecialFare | NUM10 | Y | | | N | 200 | Special Fare |

#### 3.1.1.7.3 Processing Rules

None

#### 3.1.1.7.4 Loader Behavior

Insert/update

#### 3.1.1.7.5 Schema

The content of schema can be found at /config/interface/xsd/specialfare.xsd file.

#### 3.1.1.7.6 Example

```
<?xml version="1.0"?>
<SpecialFareData>
  <MFS>
    <MarketID>1</MarketID>
    <TravelOrgn>JFK</TravelOrgn>
    <TravelDestn>LHR</TravelDestn>
    <EffectDate>20000501</EffectDate>
    <ImplDate>20170501</ImplDate>
    <DeactDate>20171231</DeactDate>
    <DiscontDate>20171231</DiscontDate>
    <Compartment>Y</Compartment>
    <SpecialFare>200</SpecialFare>
  </MFS>
  <MFS>
    <MarketID>2</MarketID>
    <TravelOrgn>LHR</TravelOrgn>
    <TravelDestn>DEL</TravelDestn>
    <EffectDate>20170501</EffectDate>
    <ImplDate>20170501</ImplDate>
    <DeactDate>20171231</DeactDate>
    <DiscontDate>20171231</DiscontDate>
    <Compartment>Y</Compartment>
    <SpecialFare>325</SpecialFare>
  </MFS>
</SpecialFareData>
```

### *3.1.1.8 Marketing Flight Fare*

#### 3.1.1.8.1 Description

The marketing flight fare table defines the marketing flight (non-od control) fare for the group system evaluation.

#### 3.1.1.8.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Origin | CITY | Y | | | Y | BOM | Origin shall be an airport code |
| Destination | CITY | Y | | | Y | LHR | Destination shall be an airport code. |
| Compartment | CHAR 02 | Y | | | Y | Y | Requested Compartment.<br>This field must be a specific compartment. |
| Carrier | CARRIER | Y | | | Y | BA | Carrier Code. |
| FromFlight | FLIGHT | Y | | | Y | 00001 | The *FLIGHT* field type is the same as the CHAR field type with a maximum length of five (5) characters. This field contains flight numbers. This field may contain only numeric characters |
| ToFlight | FLIGHT | Y | | | Y | 00200 | The *FLIGHT* field type is the same as the CHAR field type with a maximum length of five (5) characters. This field contains flight numbers. This field may contain only numeric characters |
| StartDate | DATE | Y | | | Y | 20040801 | The start date for which this fare will apply to the flight. |
| EndDate | DATE | Y | | | Y | 20040801 | The end date for which this fare will apply to the flight. . |
| Frequency | FLAG 07 | Y | | | Y | 0000011 | The first digit must start with Monday. The example shows Saturday and Sunday are the weekend.<br>Provide 1111111 if frequency is not a sensitive factor. |
| Value | NUMBER 10 | Y | | | N | 100 | fare |

#### 3.1.1.8.3 Processing Rules

None

#### 3.1.1.8.4 Loader Behavior

Insert/update

#### 3.1.1.8.5 Schema

The content of schema can be found at /config/interface/xsd/mktfligtfare.xsd file

#### 3.1.1.8.6 Example

```
<?xml version="1.0"?>
<MktFltFareData>
 <MFF>
  <Origin>PVG</Origin>
  <Destination>FRA</Destination>
```

```
   <Compartment>M</Compartment>
   <Carrier>LH</Carrier>
  <FromFlight>00100</FromFlight>
   <ToFlight>99999</ToFlight>
   <StartDate>20010101</StartDate>
   <EndDate>20090101</EndDate>
   <Frequency>1000010</Frequency>
   <Value>138</Value>
  </MFF>
</MktFltFareData>
```

### *3.1.1.9 Market Matrix*

#### 3.1.1.9.1 Description

Weekend can be defined differently on different market.

#### 3.1.1.9.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| MarketID | NUM 03 | Y | | | Y | 1 | Market ID |
| WeekendFrequency | FLAG 07 | Y | | | N | 0000011 | The first digit must start with Monday. The example shows Saturday and Sunday are the weekend. |

#### 3.1.1.9.3 Processing Rules

None

#### 3.1.1.9.4 Loader Behavior

Insert/Update

#### 3.1.1.9.5 Schema

The content of schema can be found at /config/interface/xsd/mktmtrix.xsd file.

#### 3.1.1.9.6 Example

```
<?xml version="1.0"?>
<MarketMatrixData>
  <MMG>
    <MarketID>1</MarketID>
    <WeekendFrequency>0000011</WeekendFrequency>
  </MMG>
  <MMG>
    <MarketID>2</MarketID>
    <WeekendFrequency>1100000</WeekendFrequency>
  </MMG>
</MarketMatrixData>
```

### *3.1.1.10 Market Matrix Date Range*

#### 3.1.1.10.1 Description

Specify effective date range values for all fare matric ids (3.3.1.11) under each market. The market matrix data 3.3.1.9 must exist before the related date range data can be populated.

#### 3.1.1.10.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| MarketID | NUM 03 | Y | | | Y | 1 | Market ID |
| MatrixID | NUM 03 | Y | | | N | 1 | Matrix ID |
| EffectDate | DATE | Y | | | Y | 20090501 | Effective Date for the matrix id |
| DiscontDate | DATE | Y | | | Y | 20101231 | Discontinue Date for the matrix id. |

#### 3.1.1.10.3 Processing Rules

None

#### 3.1.1.10.4 Loader Behavior

Insert/Update

#### 3.1.1.10.5 Schema

The content of schema can be found at /config/interface/xsd/mktmtrixrange.xsd file.

#### 3.1.1.10.6 Example

```
<?xml version="1.0"?>
<MarketMatrixDateRangeData>
  <MMD>
    <MarketID>1</MarketID>
    <MatrixID>1</MatrixID>
    <EffectDate>20080101</EffectDate>
    <DiscontDate>20090101</DiscontDate>
  </MMD>
  <MMD>
    <MarketID>1</MarketID>
    <MatrixID>2</MatrixID>
    <EffectDate>20090101</EffectDate>
    <DiscontDate>20100101</DiscontDate>
  </MMD>
</MarketMatrixDateRangeData>
```

### *3.1.1.11 Fare Matrix*

#### 3.1.1.11.1 Description

This filed is to populate the market fare adjustment based on the ratio of the marginal fare to the market fare. If the ratio value is within this range, the additional percentage adjustment will apply to the market fare.

#### 3.1.1.11.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| MatrixID | NUM 03 | Y | | | Y | 1 | Matrix ID |
| LowBound | NUM 03 | Y | | | N | 26 | Low bound percentage of marginal fare to market fare |
| HighBound | NUM 03 | Y | | | N | 50 | High bound percentage of marginal fare to market fare |
| AdjPercentage | NUM 03 | Y | | | N | 15 | Adjustment percentage applies to the market fare |

#### 3.1.1.11.3 Processing Rules

None

#### 3.1.1.11.4 Loader Behavior

Insert/Update

#### 3.1.1.11.5 Schema

The content of schema can be found at /config/interface/xsd/faremtrix.xsd file.

#### 3.1.1.11.6 Example

```
<?xml version=”1.0”?>
<MarketFareAdjustmentData>
  <MFM>
    <MatrixID>1</MatrixID>
    <LowBound>0</LowBound>
    <HighBound>25</HighBound>
    <AdjPercentage>60</AdjPercentage>
  </MFM>
  <MFM>
    <MatrixID>1</MatrixID>
    <LowBound>26</LowBound>
    <HighBound>50</HighBound>
    <AdjPercentage>75</AdjPercentage>
  </MFM>
  <MFM>
    <MatrixID>1</MatrixID>
    <LowBound>51</LowBound>
```

```
    <HighBound>75</HighBound>
    <AdjPercentage>80</AdjPercentage>
  </MFM>
  <MFM>
    <MatrixID>1</MatrixID>
    <LowBound>76</LowBound>
    <HighBound>90</HighBound>
    <AdjPercentage>95</AdjPercentage>
  </MFM>
</MarketFareAdjustmentData>
```

### *3.1.1.12 Exchange Rate*

#### 3.1.1.12.1 Description

The exchange rate table defines the currency and conversion rate for the quoted fare. GRMS will utilize this table and calculate the quoted fare for Adhoc requests. Currently, this logic is only applied to certain carriers.

#### 3.1.1.12.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| LCU | CHAR 03 | Y | | | Y | USD | Local Currency Code. |
| ExchangeRate | FLT 11.5 | Y | | | Y | 1.5 | Rate-of-Exchange.<br>Quoted Fare = Local Fare * Exchaneg Rate |
| EffDate | DATE | Y | | | Y | 20060101 | This field contains the date starting from which the Exchange Rate is valid<br>Format: YYYYMMDD |
| DisContDate | DATE | Y | | | Y | 200600331 | This field contains the last departure date the Exchange Rate is valid.<br>Format: YYYYMMDD |

#### 3.1.1.12.3 Processing Rule

None

#### 3.1.1.12.4 Loader Behavior

Insert/Update

#### 3.1.1.12.5 Schema

The content of schema can be found at /config/interface/xsd/exchange.xsd file.

#### 3.1.1.12.6 Example

```
<ExchangeRateData>
      <EXR>
      <LCU>TWD</LCU>
      <ExchangeRate>0.03</ExchangeRate>
```

```
      <EffDate>20060101</EffDate>
      <DisContDate>20060131</DisContDate>
      </EXR>

      <EXR>
      <LCU>USD</LCU>
      <ExchangeRate>1.000</ExchangeRate>
      <EffDate>20060101</EffDate>
      <DisContDate>20060131</DisContDate>
      </EXR>
</ExchangeRateData>
```

### *3.1.1.13 Peak/Off Peak period*

#### 3.1.1.13.1 Description

This feature will allow the administrators to configure the application so that specific data based on the departure date ranges can be used to calculate the materialization for a request. The first departure date of the request will be used to determine the seasonal period.

#### 3.1.1.13.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Orgn | CITY | Y | | | Y | JFK | Name of the origin city for this market. If it is a default market, provide '*' as the orgn. |
| Dstn | CITY | Y | | | Y | LHR | Name of the destination city for this market. If it is a default market, provide '*' as the dstn. |
| CountryPOS | CHAR 05 | N | | | Y | US | Country Point of Sale. If it is a default market, provide '*' as the country pos. |
| Desc | CHAR 30 | N | | | N | | Text. Don't pass this field if there is no value. |
| <SuperPeak> | | N | | | Y | | Provide the SuperPeak if there is any data defined. |
| <Peak> | | N | | | Y | | Provide the Peak if there is any data defined. |
| <OffPeak> | | N | | | Y | | Provide the OffPeak if there is any data defined. |

##### *3.1.1.13.2.1 Super Peak*

RecordSet: SuperPeak

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| StartDate | DATE | Y | | | Y | 20060101 | Format: YYYYMMDD |
| EndDate | DATE | Y | | | N | 20060101 | Format: YYYYMMDD |

##### *3.1.1.13.2.2 Peak*

RecordSet: Peak

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| StartDate | DATE | Y | | | Y | 20060101 | Format: YYYYMMDD |
| EndDate | DATE | Y | | | N | 20060101 | Format: YYYYMMDD |

#### *3.1.1.13.2.3 Off Peak*

RecordSet: OffPeak

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| StartDate | DATE | Y | | | Y | 20060101 | Format: YYYYMMDD |
| EndDate | DATE | Y | | | N | 20060101 | Format: YYYYMMDD |

### 3.1.1.13.3 Process Rule

Any overlapping of date range across different peak definition will be considered invalid.

### 3.1.1.13.4 Loader Behavior

Insert

### 3.1.1.13.5 Schema

The content of schema can be found at /config/interface/xsd/peak.xsd file.

### 3.1.1.13.6 Example

```
<?xml version="1.0" ?>
<PeakData>
<MKT>
    <Orgn>LAX</Orgn>
    <Dstn>TPE</Dstn>
    <CountryPOS>US</CountryPOS>
    <Desc>This is a pacific market</Desc>
    <SuperPeak>
        <StartDate>20040110</StartDate>
        <EndDate>20040215</EndDate>
</SuperPeak>

<SuperPeak>
    <StartDate>20050110</StartDate>
    <EndDate>20050215</EndDate>
</SuperPeak>

<SuperPeak>
    <StartDate>20060110</StartDate>
    <EndDate>20060215</EndDate>
</SuperPeak>

<Peak>
    <StartDate>20040615</StartDate>
    <EndDate>20040715</EndDate>
</Peak>

<Peak>
    <StartDate>20050615</StartDate>
    <EndDate>20050715</EndDate>
</Peak>
```

```
<Peak>
    <StartDate>20060615</StartDate>
    <EndDate>20060715</EndDate>
</Peak>

<OffPeak>
    <StartDate>20040315</StartDate>
    <EndDate>20040430</EndDate>
</OffPeak>

<OffPeak>
    <StartDate>20050315</StartDate>
    <EndDate>20050430</EndDate>
</OffPeak>

<OffPeak>
    <StartDate>20060315</StartDate>
    <EndDate>20060430</EndDate>
</OffPeak>
</MKT>
<MKT>
    <Orgn>*</Orgn>
    <Dstn>*</Dstn>
    <CountryPOS>*</CountryPOS>
    <Desc>This is a default market</Desc>
    <SuperPeak>
        <StartDate>20040716</StartDate>
        <EndDate>20040816</EndDate>
  </SuperPeak>

<SuperPeak>
    <StartDate>20050716</StartDate>
    <EndDate>20050816</EndDate>
  </SuperPeak>

<SuperPeak>
    <StartDate>20060716</StartDate>
    <EndDate>20060816</EndDate>
  </SuperPeak>

<Peak>
    <StartDate>20041120</StartDate>
    <EndDate>20041130</EndDate>
  </Peak>

<Peak>
    <StartDate>20051120</StartDate>
    <EndDate>20051130</EndDate>
  </Peak>

<Peak>
    <StartDate>20061120</StartDate>
    <EndDate>20061130</EndDate>
  </Peak>

<OffPeak>
    <StartDate>20040216</StartDate>
    <EndDate>20040310</EndDate>
  </OffPeak>
```

```
<OffPeak>
    <StartDate>20050216</StartDate>
    <EndDate>20050310</EndDate>
  </OffPeak>

<OffPeak>
    <StartDate>20060216</StartDate>
    <EndDate>20060310</EndDate>
  </OffPeak>
</MKT>
</PeakData>
```

### *3.1.1.14 Traffic Area*

#### 3.1.1.14.1 Description

The Traffic Area table defines the various customer configured traffic areas.

#### 3.1.1.14.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| TrafficArea | CHAR 03 | Y | | | Y | NA | Traffic Area |
| Description | CHAR 30 | N | | | N | North America | Traffic Area Description |

#### 3.1.1.14.3 Process Rule

None

#### 3.1.1.14.4 Loader Behavior

Insert/Update

#### 3.1.1.14.5 Schema

The content of schema can be found at /config/interface/xsd/trafficarea.xsd file.

#### 3.1.1.14.6 Example

```
<?xml version="1.0"?>
<TrafficAreaData>
    <TAD>
        <TrafficArea>NA</TrafficArea>
        <Description>NORTH AMERICA</TrafficArea>
    </TAD>
    <TAD>
        <TrafficArea>SA</TrafficArea>
        <Description>SOUTH AMERICA</TrafficArea>
    </TAD>
</TrafficAreaData>
```

### *3.1.1.15 Region*

#### 3.1.1.15.1 Description

The Region table defines the various customer configured regions.

#### 3.1.1.15.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Region | CHAR 05 | Y | | | Y | NYN | Region |
| Description | CHAR 30 | N | | | N | New York City | Region Description |

#### 3.1.1.15.3 Process Rule

None

#### 3.1.1.15.4 Loader Behavior

Insert/Update

#### 3.1.1.15.5 Schema

The content of schema can be found at /config/interface/xsd/region.xsd file.

#### 3.1.1.15.6 Example

```
<?xml version=”1.0”?>
<RegionData>
    <RGD>
        <Region>NYC</Region>
        <Description>NEW YORK CITY</TrafficArea>
    </RGD>
    <RGD>
        <Region>LON</Region>
        <Description>LONDON</TrafficArea>
    </RGD>
</RegionData>
```

### *3.1.1.16 Country POS*

#### 3.1.1.16.1 Description

The Country POS table defines the various customer configured Country Point of Sales.

#### 3.1.1.16.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| CountryPOS | CHAR 05 | Y | | | Y | USA | Country |
| Description | CHAR 30 | N | | | N | UNITED STATES | Country Description |
| Region | CHAR 05 | N | | | N | NAM | Region Code |
| TrafficArea | CHAR 03 | N | | | N | NA | Traffic Area |
| NameProtectPercent | NUM 03 | N | | | N | 80 | Minimum name protection percentage value used by the PNRLoad booking profile adjustment. |

#### 3.1.1.16.3 Process Rule

None

#### 3.1.1.16.4 Loader Behavior

Insert/Update

#### 3.1.1.16.5 Schema

The content of schema can be found at /config/interface/xsd/country.xsd file.

#### 3.1.1.16.6 Example

```
<?xml version="1.0"?>
<CountryData>
  <CTD>
    <CountryPOS>US</CountryPOS>
    <Description>UNITED STATES</Description>
    <Region>NA</Region>
    <TrafficArea>XX</TrafficArea>
    <NameProtectPercent>90</NameProtectPercent>
  </CTD>
  <CTD>
    <CountryPOS>MX</CountryPOS>
    <Description>MEXICO</Description>
    <Region>NA</Region>
    <TrafficArea>YY</TrafficArea>
    <NameProtectPercent>80</NameProtectPercent>
  </CTD>
</CountryData>
```

### *3.1.1.17 OAL Fare*

#### 3.1.1.17.1 Description

The OAL fare table defines the OAL fare for the group system evaluation

#### 3.1.1.17.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Origin | CITY | Y | | | | BOM | Origin shall be an airport code |
| Destination | CITY | Y | | | | LHR | Destination shall be an airport code. |
| PointOfSale | CHAR 10 | Y | | | | IN | Point of Sale. This field may be populated by a specific POS value or non–specific POS (*) |
| ClassCode | SYMBOL 02 | Y | | | | A | Fare Class.<br>This field can be a specific class or class nonspecific (*). |
| Compartment | CHAR 02 | Y | | | | Y | Requested Compartment.<br>This field must be a specific compartment. |
| Carrier | CARRIER | Y | | | | BA | Interline Carrier Code.<br>This field may be populated by either a specific carrier code (BA). |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| ValueType | NUM 01 | Y | | | | 1 | The Value Type field indicates the method of adjustment.<br>Valid values are 1 for factor (multiply) or 2 for actual value passed from the record<br>For example, a Value Type of 1 and Value of 0.8 would multiply the Market Fare by 0.8. This would be the adjusted fare.<br>A Value Type of 2 and value of 200 would indicate that the OAL fare is 200. |
| Value | FLOAT | Y | | | | 0.8 | The interline adjustment value for this od/pos/class |
| StartDate | DATE | Y | | | | 20040801 | The first departure date for which this interline adjustment become effective.<br>If the same carrier/O&D/compartment/class/pos have different values of start date, then the closest start date to the travel departure date will be used.<br>For example, one start date has value 20080101 and the other has value 20080601. If the travel 43eparture date is 20080815, then the record with start date value 20080601 will be used for the OAL calculation. |
| EndDate | DATE | N | | | | 20040801 | The last departure date for which this interline adjustment is effective.<br>If this value is not provided, then the sustem will consider the record is valid without any expiration date. |
| TripOrigin | CITY | Y | | | | BOM | Airport code of direction origin |
| TripDestination | CITY | Y | | | | LHR | Airport code of direction destination |
| StartTime | CHAR04 | Y | | | | 1232 | Start of time range in 24 hour format |
| EndTime | CHAR04 | N | | | | 1756 | End of time range in 24 hour format<br>If not provided, G6 will populate 2359 by default |
| PaxMin | NUM 03 | Y | | | | 5 | Minimum passengers for this fare |
| PaxMax | NUM 03 | N | | | | 100 | Maximum passgengers for this fare<br>If not provided, G6 willl populate the default value 999. |
| Frequency | FLAG 07 | Y | | | | 0000011 | The first digit must start with Monday. The example shows Saturday and Sunday are the weekend.<br>Provide 1111111 if frequency is not a sensitive factor. |

### 3.1.1.17.3 Process Rule

None

#### 3.1.1.17.4 Loader Behavior

Insert/Update

#### 3.1.1.17.5 Schema

The content of schema can be found at /config/interface/xsd/oal.xsd file.

#### 3.1.1.17.6 Example 1

```
<?xml version=”1.0”?>
<OALFareData>
  <FVI>
    <Origin>BOM</Origin>
    <Destination>LHR</Destination>
    <PointOfSale>IN</PointOfSale>
    <ClassCode>Y</ClassCode>
    <Compartment>Y</Compartment>
    <Carrier>ZZ</Carrier>
    <ValueType>1</ValueType>
    <Value>0.8</Value>
    <StartDate>20080101</StartDate>
    <EndDate>20081231</EndDate>
    <TripOrigin>IAH</TripOrigin>
    <TripDestination>BEJ</TripDestination>
    <StartTime>1235</StartTime>
    <PaxMin>5</PaxMin>
    < Frequency>0110000</Frequency>
  </FVI>
</OALFareData>
```

#### 3.1.1.17.7 Example 2

```
<?xml version=”1.0”?>
<OALFareData>
  <FVI>
    <Origin>BOM</Origin>
    <Destination>LHR</Destination>
    <PointOfSale>IN</PointOfSale>
    <ClassCode>Y</ClassCode>
    <Compartment>Y</Compartment>
    <Carrier>ZZ</Carrier>
    <ValueType>1</ValueType>
    <Value>0.8</Value>
    <StartDate>20080101</StartDate>
    <EndDate>20081231</EndDate>
    <TripOrigin>IAH</TripOrigin>
    <TripDestination>BEJ</TripDestination>
    <StartTime>1235</StartTime>
    <EndTime>1756</EndTime>
    <PaxMin>5</PaxMin>
    <PaxMax>50</PaxMax>
    < Frequency>0110000</Frequency>
  </FVI>
</OALFareData>
```

### *3.1.1.18 Deposit Deadline*

#### 3.1.1.18.1 Description

The deposit deadline table defines the date in which the deposit is due in order to prevent the PNR from being cancelled.

#### 3.1.1.18.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| CountryPos | CHAR 02 | Y | | | | QR | Country point of sale |
| RequestType | CHAR 02 | Y | | | | A | Request Type can be either Series (S), Adhoc (A) or Both (B) |
| BookingRangeFrom | NUM 10 | Y | | | | 360 | Maximum number of days prior to the first departure date up to which a deposit is accepted. This is the first part of the Booking Range. |
| BookingRangeTo | NUM 10 | Y | | | | 61 | Minimum number of days prior to the first departure date up to which a deposit is accepted. This is the second part of the Booking Range. |
| Period | SYMBOL 02 | Y | | | | 1 | Identifies whether the departure date is within a Peak or Off Peak period.<br>There are 4 types of records that can be defined in the peak/off-peak table.<br>1) Super Peak – Considered Peak travel period<br>2) Peak – considered Peak travel Period<br>3) Off Peak – Off Peak travel Period<br>4) Default – Not defined as Super Peak, Peak or Off Peak |
| DepositDeadline | NUM 10 | Y | | | | 1 | The number of days that will be used to determine the deposit deadline date. |
| DAC | NUM 01 | Y | | | | 1 | Will be either 1 for "**D**ays **A**fter **C**onfirmation", or 0 for "Days **P**rior **T**o **D**eparture". |

#### 3.1.1.18.3 Process Rule

None

#### 3.1.1.18.4 Loader Behavior

Insert/Update

#### 3.1.1.18.5 Schema

The content of schema can be found at /config/interface/xsd/depdeadline.xsd file.

#### 3.1.1.18.6 Example 1

```
<?xml version="1.0"?>
<DepositDeadlineData>
  <DDL>
```

```
    <CountryPos>QR</CountryPos>
    <RequestType>A</RequestType>
    <BookingRangeFrom>360</BookingRangeFrom>
    <BookingRangeTo>61</BookingRangeTo>
    <PeriodId>2</PeriodId>
    <DepositDeadline>21</DepositDeadline>
    <DAC>0</DAC >
  </DDL>
</DepositDeadlineData>
```

#### 3.1.1.18.7 Example 2

```
<?xml version="1.0"?>
<DepositDeadlineData>
  <DDL>
    <CountryPos>ES</CountryPos>
    <RequestType>S</RequestType>
    <BookingRangeFrom>60</BookingRangeFrom>
    <BookingRangeTo>46</BookingRangeTo>
    <PeriodId>1</PeriodId>
    <DepositDeadline>23</DepositDeadline>
    <DAC>1</DAC>
  </DDL>
</DepositDeadlineData>
```

### *3.1.1.19 Name Firming Deadline*

#### 3.1.1.19.1 Description

The name firming deadline table defines the name firming percentage and the date in which the name firming is due in order to prevent the PNR from being cancelled.

#### 3.1.1.19.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| CountryPos | CHAR 02 | Y | | | | QR | Country point of sale |
| BookingRangeFrom | NUM 10 | Y | | | | 360 | Maximum number of days prior to the first departure date up to which name firming is valid. This is the first part of the Booking Range. |
| BookingRangeTo | NUM 10 | Y | | | | 61 | Minimum number of days prior to the first departure date up to which name firming is valid. This is the second part of the Booking Range. |
| NamesPercentage | NUM 03 | Y | | | | 85 | The percentage the name firming must be completed in the determined date. |
| DeadlineDays | NUM 10 | Y | | | | 1 | The number of days that will be used to determine the name firming deadline. |
| DAC | NUM 01 | Y | | | | 1 | 1 for Days after confirmation, 0 for Days Prior to Departure |

#### 3.1.1.19.3 Process Rule

None

#### 3.1.1.19.4 Loader Behavior

Insert/Update

#### 3.1.1.19.5 Schema

The content of schema can be found at /config/interface/xsd/namefirmingdeadline.xsd file.

#### 3.1.1.19.6 Example 1

```
<?xml version="1.0"?>
<NameFirmingDeadlineData>
  <NFD>
    <CountryPos>QR</CountryPos>
    <BookingRangeFrom>360</BookingRangeFrom>
    <BookingRangeTo>61</BookingRangeTo>
    <NamesPercentage>66</NamesPercentage>
    <DeadlineDays>32</DeadlineDays>
    <DAC>0</DAC >
  </NFD>
</NameFirmingDeadlineData>
```

#### 3.1.1.19.7 Example 2

```
<?xml version="1.0"?>
<NameFirmingDeadlineData>
  <NFD>
    <CountryPos>ES</CountryPos>
    <BookingRangeFrom>232</BookingRangeFrom>
    <BookingRangeTo>55</BookingRangeTo>
    <NamesPercentage>85</NamesPercentage>
    <DeadlineDays>42</DeadlineDays>
    <DAC>1</DAC >
  </NFD>
</NameFirmingDeadlineData>
```

### *3.1.1.20 Flight Group Capacity Control*

#### 3.1.1.20.1 Description

The flight group capacity control table defines the percentage limit a group is allowed to book on a certain flight. This percentage number will apply to the adjusted capacity of the flight to derive the maximum group bookings allowed on the flight.

#### 3.1.1.20.2 Data Fields

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| CrrCode | CARRIER | Y | | | Y | ZZ | Carrier Code. Use * if the carrier code is not specified. |
| FlightNum | FLIGHT 05 | Y | | | Y | 00441 | Flight number. Use * if the flight number is not specified. |
| SegOrgn | CITY | Y | | | Y | IAH | Travel Segment Origin. Group boarding airport point. |
| SegDstn | CITY | Y | | | Y | FRA | Travel Segment Destination. Group off-boarding airport point. |
| CmpCode | SYMBOL 02 | Y | | | Y | Y | Compartment Code. |
| EffectDate | DATE | Y | | | Y | 20130501 | Effective Date |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DiscontDate | DATE | Y | | | Y | 20131231 | Discontinue Date |
| CapcityPercent | NUMBER 03 | Y | | | N | 70 | Maximum group book percentage. |

#### 3.1.1.20.3 Process Rule

None

#### 3.1.1.20.4 Loader Behavior

Insert/Update

#### 3.1.1.20.5 Schema

The content of schema can be found at /config/interface/xsd/fltgroupcapacity.xsd file.

#### 3.1.1.20.6 Example

```
<?xml version="1.0"?>
<FlightCapacityData>
  <FCG>
    <CrrCode>XX</CrrCode>
    <FlightNum>01234</FlightNum>
    <SegOrgn>IAH</SegOrgn>
    <SegDstn>LAX</SegDstn>
    <CmpCode>Y</CmpCode>
    <EffectDate>20130201</EffectDate>
    <DiscontDate>20130501</DiscontDate>
    <CapacityPercent>70</CapacityPercent>
  </FCG>
  <FCG>
    <CrrCode>XX</CrrCode>
    <FlightNum>05678</FlightNum>
    <SegOrgn>IAH</SegOrgn>
    <SegDstn>TPE</SegDstn>
    <CmpCode>Y</CmpCode>
    <EffectDate>20130601</EffectDate>
    <DiscontDate>20130815</DiscontDate>
    <CapacityPercent>65</CapacityPercent>
  </FCG>
  <FCG>
    <CrrCode>*</CrrCode>
    <FlightNum>*</FlightNum>
    <SegOrgn>IAH</SegOrgn>
```

```
    <SegDstn>FRA</SegDstn>
    <CmpCode>Y</CmpCode>
    <EffectDate>201301115</EffectDate>
    <DiscontDate>20131231</DiscontDate>
    <CapacityPercent>70</CapacityPercent>
  </FCG>
</FlightCapacityData>
```

### 3.1.2 Transactional Data

Transactional data sets are those that change more frequently. The PROS system will maintain as much transactional history as necessary. Records in these data sets must be only those which contain any changes.

| | DataSet Name | RecordSet Name | Record Set Description |
|---|---|---|---|
| 1 | RequestLoad | RHG | Ad hoc Request load |
| 2 | PnrLoad | PnrData (LoadingDate ="20051031") | PNR nightly load. We set the attribute at PnrData.<br>This LoadingDate is used to store the date of Reservation System when PNR Load file was created. It is considered as the date when the Reservation System has the data snapshot with any PNR updates. |
| 3 | AcceptLoad | AHG | Acceptance Request load |
| 3 | SeriesLoad | SHG | Series Request load |

#### *3.1.2.1 Ad hoc Request Load*

##### 3.1.2.1.1 Description

Group requests collected from Airlines' Mask must be loaded into the PROS Group System. Airlines must provide PROS Group System with a file containing general information for header, direction, date range, and miscellaneous free form text. PROS Group System will use information from the Mask to create a PNR for each group request, if it confirmed.

Although it is possible to include multiple requests in a request file, because of the real-time nature of the data exchange, it is advisable to create a new file for each Group Request. This will prevent any issues that may be cause by file locking while the request file is being processed.

The PROS Group System will accept whatever data is passed to it. If the request file has the necessary data (correct or incorrect) and is formatted correctly, it will be loaded. In the case that the corruption was to the formatting or so severe that the request is not loadable, it will not be loaded. It is the Airlines' responsibility to create the clean load file and put it in the data directory.

Note: Please review the "Data Field" column to create xml or Json (for rest services).

### 3.1.2.1.2 Data Fields

#### *3.1.2.1.2.1 Request Header*

This record contains general request level information. There will be one RHG record for each request.

RecordSet: RHG (for xml) and "rhg" (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| RqstID (XML)<br>rqstID (JSON) | CHAR 10 | Y | | | | Group433 | Unique Reference Number used for identifying a Group Request application. |
| OutstationID (XML)<br>outstationID (JSON) | CHAR 09 | Y | | | | OUT11 | Airlines Outstation/Sales office where request originates from. |
| RequestDate (XML)<br>requestDate (JSON) | DATE | Y | | | | 20050930 | Date group booking request submitted to G6.<br>Format: YYYYMMDD |
| RequestTime (XML)<br>requestTime (JSON) | TIME 04 | Y | | | | 1130 | Time group booking request submitted to G6.<br>Format: 0000 – 2359 |
| GroupName (XML)<br>groupName (JSON) | CHAR 50 | Y | | | | U OF H GROUP | Name of the group for use in PNR Creation. |
| ProgramName (XML)<br>programName (JSON) | CHAR 50 | N | | | | Summer Travel | Additional identification for Group. Don't pass this field if there is no value. |
| TourName (XML)<br>tourName (JSON) | CHAR 50 | N | | | | Excursion | Additional identification for Group. Don't pass this field if there is no value. |
| CustomerType (XML)<br>customerType (JSON) | NUM 10 | Y | Customer type | | | 3 | Number used to identify category type (e.g.: Leisure, Cruise, Sports, etc.) |
| AgencyIata (XML)<br>agencyIata (JSON) | CHAR 10 | N | Agency | | | A987654321 | GDS or other Airlines IATA reference. Don't pass this field if there is no value. |
| OfficeCode (XML)<br>officeCode (JSON) | CHAR 10 | N | Agency | | | FRALH0860 | Office Code.<br>Either IATA or Office Code must exist. Don't pass this field if there is no value. |
| AgencyName (XML)<br>agencyName (JSON) | CHAR 50 | N | Agency | | | Discount Travelers | GDS or other Airlines Name. Don't pass this field if there is no value. |
| GroupType (XML)<br>groupType (JSON) | CHAR 02 | Y | | | | A | A = Ad hoc<br>S = Series |
| CityPOS (XML)<br>cityPOS (JSON) | CITY | Y | | | | IAH | Departure Airport for the City where the request originated.<br>In case of new Agency IATA, this field will populate the City POS field in the Agency table in G6. Don't pass this field if there is no value. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DistPOS (XML)<br>distPOS (JSON) | CITY | Y | | | | US | Country or Airport code of the City where request originated.<br>In case of new Agency IATA, this field will populate the District POS field in the Agency table in G6. Don't pass this field if there is no value. |
| MarketOrg (XML)<br>marketOrg (JSON) | CITY | Y | Market O&D | | | IAH | Default to first direction of origin. |
| MarketDstn (XML)<br>marketDstn (JSON) | CITY | Y | Market O&D | | | FRA | Default to first direction of destination. |
| CommPnct (XML)<br>commPnct (JSON) | NUM 02 | Y | | | | 0 | Agent commission percentage.<br>Must be a whole number. |
| SeatsRqstd (XML)<br>seatsRqstd (JSON) | NUM 03 | Y | | | | 20 | |
| SeatsPasses (XML)<br>seatsPasses (JSON) | NUM 03 | Y | | | | 0 | Non-revenue tickets. |
| MinSeatRqstd (XML)<br>minSeatRqstd (JSON) | NUM 03 | N | | | | 20 | This value will be used only if<br>Passenger negotiation is set (NegoFlag )<br>Its value can only be smaller than the seats requested and greater than 0.<br>If this value is not passed and the NegoFlag is set to Passenger (4), the request will be rejected. |
| QuotedFare (XML)<br>quotedFare (JSON) | NUM 10 | N | | | | 500 | Price the group is willing to pay per passenger. This fare is compared with the Marginal Fare to determine if the group should be accepted. It must be entered in the Mask by the sales person. However, if the business side determines that group requests can be accepted without a quoted fare, it may be left empty.<br>The quoted fare must be in the same currency of the supporting Revenue Management System (i.e. PROS).<br>Please check on the 3.1.3.8 on maximum positive number 2147483647. |
| LCU (XML)<br>lcu (JSON) | CHAR 03 | N | | | | USD | Currency code of the Airlines Outstation or RES Office (i.e. LAX office = USD, LON office = GBP). Don't pass this field if there is no value. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| LocalFare (XML)<br><br>localFare (JSON) | NUM 10 | N | | | | 500 | Fare being charged per person in Local Currency Code as defined above. Don't pass this field if there is no value.<br>Please check on the 3.1.3.8 on maximum positive number 2147483647. |
| WorkQueueID (XML)<br><br>workQueueID (JSON) | NUM 03 | N | | | | 0 | Numerical ID of Work Team request will be assigned to.<br>0 = Central (Default). Don't pass this field if there is no value. |
| PreferredFlightFlag (XML)<br><br>preferredFlightFlag (JSON) | FLAG 01 | Y | | | | 0 | Flag to indicate that CCITIN must return from RES options with requested flight only.<br>1 = Yes<br>0 = No (Default) |
| PnrGenFlag (XML)<br><br>pnrGenFlag (JSON) | FLAG 01 | N | | | | 1 | 1 = Yes (Default)<br>0 = No<br>Flag for G6 to create a PNR should the offered flights/fare be accepted.<br>NOTE:<br>If the PNR Generation Flag is set to no, it is not possible to generate a PNR through the system. The only path to follow is through the negotiation facility. If a request is sent with PNR set to no and an option is sent and accepted by the sales agent, it is clear the agent is ready for a PNR to be generated by the acceptance of the negotiation. In this case, the acceptance message is loaded back to PROS Group System and the user will generate a PNR from the Group Shell/PNR Editor. It is important to remember in this case, the option that is offered by the user is usually a response to the requested flight.<br>If PNR Generation flag is set to no, autopilot will still process the request but it will send options automatically back to the outstations, rather than trying to book the group. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| RequestTypeFlag (XML)<br><br>requestTypeFlag (JSON) | FLAG | Y | | | | 0 | Flag to identify different "types" of requests:<br>0 = "Regular" request (Default)<br>1 = Add-on request – to increase the number of seats originally requested<br>2 = Re-evaluation request – to modify and already booked group. |
| RPI (XML)<br><br>rpi (JSON) | NUM 03 | N | | | | | A priority index for the user to work on the requests.<br>This field is still under design.<br>Don't pass this field if there is no value. |
| ApExclusive (XML)<br><br>apExclusive (JSON) | FLAG 01 | Y | | | | 0 | Exclude the request from Autopilot execution<br>Default value is 0.<br><br>This filed will indicate if Autopilot execution will ignore this request for processing.<br>1: Ignore this request<br>0: Process this request. |
| NegoFlag (XML)<br><br>negoFlag (JSON) | NUM 01 | Y | | | | 1 | Default value is 0.<br>0: No negotiation allowed.<br>1: Flight negotiation<br>2: Fare negotiation (mainly for the preferred flight)<br>4: Passenger negotiation (mainly for the preferred flight)<br>3: Fare and Flight negotiation (mainly for the preferred flight) |
| TotalTripFare (XML)<br><br>totalTripFare (JSON) | NUM 10 | N | | | | 500 | Total fare for the trip including the OAL fare. Don't pass this field if there is no value. |
| StsReqAdjFlag (XML)<br><br>stsReqAdjFlag (JSON) | FLAG 01 | N | | | | 0 | The flag to make the utilization adjustment.<br>O: no adjustment<br>1: adjustment.<br>Don't pass this field if there is no value. |
| DealIndicatorFlag (XML)<br><br>dealIndicatorFlag (JSON) | CHAR 02 | N | | | | G | This field will indicate the Deal Type for each request (Request Header Level). It can have 3 values – G (Ad Hoc), S (Series), M (Ad Hoc). The M would indicate that the Deal will be based on a agreed upon Market Fare |
| TripType (XML)<br><br>tripType (JSON) | CHAR 02 | N | | | | RT | This field would indicate if the Group Request is for Round Trip (RT) or One Way (OW). Open Jaw Group Request will be send to G6 as a OJ. |
| SourceCRS (XML)<br>sourceCRS (JSON) | CHAR 05 | N | | | | AM | Central Reservation System Code |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| AgentURL (XML)<br>agentURL (JSON) | CHAR 128 | N | | | | | The URL to send responses back to the Agent |
| DiscountPercent (XML)<br><br>discountPercent (JSON) | CHAR 03 | N | | | | 15 | An extra discount percentage for this request associated with the City POS |
| <RFG> (XML)<br>"rfg" (JSON) | | Y | | | | | Request Direction |
| <Note> (XML)<br>"note" (JSON) | | Y | | | | | Note |
| <RHP> (XML)<br>"rhp" (JSON) | | N | | | | | Request PNR Information |

#### *3.1.2.1.2.2 Request Direction*

This record contains direction level information. There can be multiple RFG(for xml) and rfg (for JSON) records per request, one for each direction.

RecordSet: RFG (for XML) and rfg (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DirNum (XML)<br>dirNum (JSON) | NUM 02 | Y | | | | 1 | Direction number starting from 1. The outbound direction must be direction 1. |
| Orgn (XML)<br>orgn (JSON) | CITY | Y | | | | IAH | Boarding point. |
| Dstn (XML)<br>dstn (JSON) | CITY | Y | | | | FRA | Off point. |
| SumFareCase (XML)<br>sumFareCase (JSON) | CHAR02 | N | | | | CW | Sum of marginal fares for each segment.<br>If blank, use Default set up in SysParms.<br>Don't pass this field if there is no value. |
| DenyBoardingFactor (XML)<br><br>denyBoardingFactor (JSON) | CHAR 02 | N | | | | | Factor used to load marginal fare according to Fight/date demand levels.<br>If blank, use Default set up in SysParms.<br>Don't pass this field if there is no value. |
| RqstdCmp (XML)<br><br>rqstdCmp (JSON) | SYMBOL 02 | Y | Compartment Cabin mapping | | | M | Compartment code identifying requested cabin. |
| PrefClassCode (XML)<br>prefClassCode (JSON) | SYMBOL 02 | N | | | | M | Preferred directional booking Class Code.<br>Don't pass this field if there is no value. |
| PrefClassPOSCode (XML)<br>prefClassPOSCode (JSON) | CHAR 10 | N | | | | | Preferred directional booking Class POS.<br>Don't pass this field if there is no value. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Duration (XML)<br><br>duration (JSON) | NUM 03 | N | | | | 5 | Trip Duration number.<br>1. Not applicable to the first direction.<br>2. Duration or StartDate and EndDate (request range RFD) must be provided.<br><br>If startDate and endDate are provided (RFD), duration field must be empty.<br>Don't pass this field if there is no value. |
| QuotedFare (XML)<br><br>quotedFare (JSON) | NUM 10 | N | | | | | Directional Quoted. Group system will ignore the Quoted Fare information on the Request Header if the directional quoted fare is provided. Otherwise, the directional quoted fare number will be given by the Request Quoted Fare divided by the number of directions.<br>Don't pass this field if there is no value. |
| LocalFare (XML)<br>localFare (JSON) | NUM 10 | N | | | | | Local fare on the directional level. Don't pass this field if there is no value. |
| PrefDate (XML)<br>prefDate (JSON) | DATE | N | | | | | This is the preferred date set at the directional level for the market fare retrieval logic. Don't pass this field if there is no value. |
| StopOverIndicator (XML)<br><br>stopOverIndicator (JSON) | NUM 01 | N | | | | 1 | Each direction will have a Stop Over Indicator Flag. If the Flag is 1, then the Quoted Fare includes a Stop Over Cost for that direction. So G6 would need to take that into consideration during the calculation of Marginal Fares. Note that here the stop over means that the time between the 2 connecting flight is between 4 and 24 hours and the Group would like to stay in a hotel for that duration. (Do not confuse this with the way we are used to defining stop over – which usually means a new direction !). |
| StopOverCost (XML)<br>stopOverCost (JSON) | NUM 10 | N | | | | | This field will denote the Stop Over cost for that particular direction |
| <RFD> (XML)<br>"rfd" (JSON) | | Y | | | | | Request Range information |

#### *3.1.2.1.2.3*

#### *3.1.2.1.2.4 Request PNR Information*

This record contains general request level information.

Record Set: RHP (for XML) and rhp (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| PNRLocator (XML)<br>pnrLocator (JSON) | LOCATOR | N | | | | B32DT2005 0812 | This field is used fo re-evaluation and PNR request. Don't pass this field if there is no value. |
| PrimPnrLocator (XML)<br>primPnrLocator (JSON) | LOCATOR | N | | | | B32DT2005 0812 | This field is used for re-evaluation and PNR request. Don't pass this field if there is no value.<br>This field must be provided to go along with the PNRLocator. It is used in GRMS to identifiy whether the PNR is a Primary PNR or a Child PNR.<br>In case of non-split PNRs, this filed is the same as PNRLocator field.<br>In case of split PNRs, the locator of the Parent (or Master) PNR goes into this field. |
| RefPnrLocator (XML)<br>refPnrLocator (JSON) | LOCATOR | N | | | | B32DT2005 0812 | In case of add-on group requests the Reference PNR is the PNR that is being added to.<br>In the case of a request originated from a GDS PNR, the Reference PNR must be the GDS PNR. Don't pass this field if there is no value. |
| TCPNumber (XML)<br>tcpNumber (JSON) | NUM 03 | N | | | | | Total Number in Group. Don't pass this field if there is no value. |
| NumOfNames (XML)<br>numOfNames (JSON) | NUM 03 | N | | | | | Total number of Names confirmed. Don't pass this field if there is no value. |

#### *3.1.2.1.2.5 Note*

This record contains general request level information. There can be multiple Note records for each request.

RecordSet: Note (for XML) and note (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Type (XML)<br>type (JSON) | CHAR04 | Y | | | Y | CTC | The identifier for note text. There is no restriction of unique note type. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Value (XML)<br>value (JSON) | CHAR256 | Y | | | | | The text content |

#### *3.1.2.1.2.6 Request Range*

This record contains direction/date range information. There can be multiple RFD record per direction, one for each direction/date range. All RFD records date-time ranges must be mutually disjoint.

RecordSet: RFD (for XML) and rfd (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| RangeNum (XML)<br>rangeNum (JSON) | NUM 02 | Y | | | Y | 1 | Unique Identifier for current and historical PNRs in the G6 system. |
| StartDate (XML)<br>startDate (JSON) | DATE | Y | | | | 20050831 | Start Date for preferred travel.<br>Format: YYYYMMDD |
| EndDate (XML)<br>endDate (JSON) | DATE | Y | | | | 20050902 | End Date for preferred travel.<br>Format: YYYYMMDD |
| Frequency (XML)<br><br>frequency (JSON) | FLAG 07 | N | | | | 1100111 | Day(s) of the Week for preferred travel.<br>The 7 digits are treated as a flag indicator for the day of the week which the first digit represents Monday and the seventh digit represents Sunday.<br>1011001 = Mon, Wed, Thu, Sun<br>0100110 = Tue, Fri, Sat.<br>If the Frequency field contains all zeros (0000000), the frequency of RFD record will automatically calculate the entire frequency occurrence from the start date to the end date. Don't pass this field if there is no value. |
| StartTime (XML)<br>startTime (JSON) | TIME | Y | | | | 0000 | Start Time for preferred travel.<br>Format: 0000 – 2359 |
| EndTime (XML)<br>endTime (JSON) | TIME | Y | | | | 2359 | End Time for preferred travel.<br>Format: 0000 – 2359 |
| ArrDptFlag (XML)<br>arrDptFlag (JSON) | FLAG 01 | Y | | | | 0 | Indicates if Dates/Times advised above relate to Departure or Arrival.<br>1 = Arrival<br>0 = Departure (Default) |
| <RFS> (XML)<br>"rfs" (JSON) | | N | | | | | Request Segment. This field is optional.<br>This is required for the PNR requests. |

### *3.1.2.1.2.7 Request Segment*

This record contains Flight/Segment information for preferred flights. Each RFD can have multiple RFS records. If no preferred flight is specified, the RFS record can be omitted.

RecordSet: RFS (for XML) and rfs (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| SegNum (XML) segNum (JSON) | NUM 02 | Y | | | Y | 1 | The segment number within each direction will start from 1. |
| SetNum (XML) setNum (JSON) | NUM 02 | N | | | | 1 | This set number is used to group travel flights into a logical set. Different travel options can have different set of preferred flights.<br>By default, the set number will be equal to the SegNum value. The set number will start from 1.<br>For example, flight 100 HOU-TPE as set 1. Flight 101 HOU-TPE as set 2. This example shows different preferred (set) options.<br>Another example shows logical group of flights. Flight 102 HOU-TYO and flight 103 TYO-TPE will both be set 1. Flight 103 TYO-TPE will be the connecting flight for HOU-TPE trip.<br>Although G6 can support preferred flight of many sets, we recommend that this set number can be limited to 3. More distinct set numbers in the loader file may subsequently increase the processing time. |
| CrrCode (XML) crrCode (JSON) | CARRIER | Y | | | | ZZ | Carrier Code |
| FlightNum (XML) flightNum (JSON) | FLIGHT 05 | Y | | | | 441 | Flight number |
| SegOrgn (XML) segOrgn (JSON) | CITY | N | | | | IAH | Travel Segment Origin.<br>Group boarding point.. Don't pass this field if there is no value.<br>PNR requests need to provide this field |
| SegDstn (XML) segDstn (JSON) | CITY | N | | | | FRA | Travel Segment Destination.<br>Group off-boarding point.<br>PNR requests need to provide this field<br>Don't pass this field if there is no value. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| CmpCode (XML) cmpCode (JSON) | SYMBOL 02 | N | | | | M | Compartment Code. Don't pass this field if there is no value. |
| ClassCode (XML) classCode (JSON) | SYMBOL 02 | N | | | | M | Class Code. PNR requests need to provide this field Don't pass this field if there is no value. |
| DptDate (XML) dptDate (JSON) | DATE | N | | | | 20050831 | Format: YYYYMMDD. PNR requests need to provide this field Don't pass this field if there is no value. |
| DptTime (XML) dptTime (JSON) | TIME | N | | | | 1545 | Format: 0000 – 2359. PNR requests need to provide this field Don't pass this field if there is no value. |
| ArrTime (XML) arrTime (JSON) | TIME | N | | | | 0845 | Format: 0000 – 2359. PNR requests need to provide this field Don't pass this field if there is no value. |
| BkgStatus (XML) bkgStatus (JSON) | CHAR 02 | N | | | | HK | This field is required to support re-evaluation. PNR requests need to provide this field Don't pass this field if there is no value. |
| ClassPOS (XML) classPOS (JSON) | CHAR 10 | N | | | | | For Airlines who use Class POS for inventory control purposes, the contents of this field will be used to identify the POS associated with the class code defined earlier in the record (e.g. "USA" in Class POS " Y USA"). Don't pass this field if there is no value. |

#### 3.1.2.1.3 Processing Rules

None.

#### 3.1.2.1.4 Loader Behavior

Insert /Update

#### 3.1.2.1.5 Schema

The content of schema can be found at /config/interface/xsd/adhoc.xsd file.

#### 3.1.2.1.6 Example

```
<?xml version="1.0" ?>
<RequestLoad>
```

```
<RHG>
 <RqstID>Group433</RqstID>
 <OutstationID>OUT11</OutstationID>
 <RequestDate>20050930</RequestDate>
 <RequestTime>1130</RequestTime>
 <GroupName>U OF H GROUP</GroupName>
 <ProgramName>Summer Travel</ProgramName>
 <TourName>Excursion</TourName>
 <CustomerType>3</CustomerType>
 <AgencyIata>A987654321</AgencyIata>
 <OfficeCode>FRALH0860</OfficeCode>
 <AgencyName>Discount Travellers</AgencyName>
 <GroupType>A</GroupType>
 <CityPOS>IAH</CityPOS>
 <DistPOS>IAH</DistPOS>
 <MarketOrg>IAH</MarketOrg>
 <MarketDstn>FRA</MarketDstn>
 <CommPnct>0</CommPnct>
 <SeatsRqstd>20</SeatsRqstd>
 <SeatsPasses>0</SeatsPasses>
 <MinSeatRqstd>20</MinSeatRqstd>
 <QuotedFare>500</QuotedFare>
 <LCU>USD</LCU>
 <LocalFare>500</LocalFare>
 <WorkQueueID>0</WorkQueueID>
 <PreferredFlightFlag>0</PreferredFlightFlag>
 <PnrGenFlag>1</PnrGenFlag>
 <RequestTypeFlag>0</RequestTypeFlag>
 <RPI>1</RPI>
 <ApExclusive>0</ApExclusive>
 <TotalTripFare>200</TotalTripFare>
 <NegoFlag>1</NegoFlag>
 <StsReqAdjFlag>0</StsReqAdjFlag>
 <DiscountPercent>15</DiscountPercent>
 <Note>
    <Type>CTC</Type>
    <Value> Boris Walden</Value>
 </Note>
 <Note>
    <Type>RCV</Type>
    <Value> Jurgen Singer </Value>
   </Note>
<Note>
 <Type>SSR</Type>
 <Value> SSR value </Value>
 </Note>
 <Note>
    <Type>OSI</Type>
    <Value> OSI value</Value>
 </Note>
 <Note>
    <Type>IMP</Type>
    <Value>IMP value</Value>
 </Note>
 <Note>
    <Type>HLD</Type>
    <Value>HLD value</Value>
```

```
</Note>
<Note>
   <Type>STO</Type>
   <Value>This is a value to be stored in the Group Shell</Value>
</Note>
<Note>
   <Type>TXT</Type>
   <Value>This is a PNR Remark</Value>
</Note>
<RHP>
   <PNRLocator>B32DT20050812</PNRLocator>
   <PrimPnrLocator>B32DT20050812</PrimPnrLocator>
   <RefPnrLocator></RefPnrLocator>
   <TCPNumber>20</TCPNumber>
   <NumOfNames>10</NumOfNames>
</RHP>
<RFG>
   <DirNum>1</DirNum>
   <Orgn>IAH</Orgn>
   <Dstn>FRA</Dstn>
   <SumFareCase>CW</SumFareCase>
   <DenyBoardingFactor></DenyBoardingFactor>
   <RqstdCmp>M</RqstdCmp>
   <PrefClassCode>Y</PrefClassCode>
   <PrefClassPOSCode>ALL</PrefClassPOSCode>
   <Duration></Duration>
   <QuotedFare>500</QuotedFare>
   <LocalFare>500</LocalFare>
   <RFD>
      <RangeNum>1</RangeNum>
      <StartDate>20050831</StartDate>
      <EndDate>20050902</EndDate>
      <Frequency>1100111</Frequency>
      <StartTime>0000</StartTime>
      <EndTime>2359</EndTime>
      <ArrDptFlag>0</ArrDptFlag>
      <RFS>
         <SegNum>1</SegNum>
         <CrrCode>ZZ</CrrCode>
         <FlightNum>441</FlightNum>
         <SegOrgn>IAH</SegOrgn>
         <SegDstn>FRA</SegDstn>
         <CmpCode>M</CmpCode>
         <ClassCode>Y</ClassCode>
         <DptDate>20050831</DptDate>
         <DptTime>1545</DptTime>
         <ArrTime>0845</ArrTime>
         <BkgStatus>HK</BkgStatus>
         <ClassPOS>ALL</ClassPOS>
  </RFS>
  </RFD>
</RFG>
<RFG>
   <DirNum>2</DirNum>
   <Orgn>FRA</Orgn>
   <Dstn>IAH</Dstn>
   <SumFareCase>CW</SumFareCase>
```

```
      <DenyBoardingFactor></DenyBoardingFactor>
      <RqstdCmp>M</RqstdCmp>
      <PrefClassCode>Y</PrefClassCode>
      <PrefClassPOSCode></PrefClassPOSCode>
      <Duration>5</Duration>
      <QuotedFare>500</QuotedFare>
      <LocalFare>500</LocalFare>
      <RFD>
         <RangeNum>1</RangeNum>
         <StartDate>20050912</StartDate>
         <EndDate>20050915</EndDate>
         <Frequency>1100111</Frequency>
         <StartTime>0000</StartTime>
         <EndTime>2359</EndTime>
         <ArrDptFlag>0</ArrDptFlag>
         <RFS>
            <SegNum>1</SegNum>
            <CrrCode>ZZ</CrrCode>
            <FlightNum>440</FlightNum>
            <SegOrgn>FRA</SegOrgn>
            <SegDstn>IAH</SegDstn>
            <CmpCode>M</CmpCode>
            <ClassCode>Y</ClassCode>
            <DptDate>20050912</DptDate>
            <DptTime>0945</DptTime>
            <ArrTime>1545</ArrTime>
            <BkgStatus>HK</BkgStatus>
            <ClassPOS>ALL</ClassPOS>
     </RFS>
     </RFD>
  </RFG>
  </RHG>
</RequestLoad>
```

#### 3.1.2.1.7 JSON adhoc request load via adhoc rest loader endpoint

Endpoint: /group/rest/submitAdhocRequest

Exmple: http://localhost:8080/group/rest/submitAdhocRequest

#### 3.1.2.1.8 Example for JSON content for adhoc request loader

```
{
    "rhg": {
        "rqstID": "A12345",
        "outstationID": "PGKGA0101",
        "requestDate": "20171205",
        "requestTime": "0513",
        "groupName": "RETA GROUP",
        "customerType": 2,
        "agencyIata": "15394466",
        "officeCode": "PGKGA0101",
        "agencyName": "RETA GROUP",
        "groupType": "A",
        "cityPOS": "PGK",
        "distPOS": "ID",
        "marketOrg": "CGK",
        "marketDstn": "PGK",
        "commPnct": 0,
```

```
        "seatsRqstd": 11,
        "seatsPasses": 2,
        "lcu": "USD",
        "localFare": 9000.0,
        "workQueueID": 0,
        "preferredFlightFlag": 0,
        "pnrGenFlag": 1,
        "requestTypeFlag": 0,
        "rpi": 1,
        "totalTripFare": 100.0,
        "negoFlag": 0,
        "stsReqAdjFlag": 0,
        "note": [{
            "type": "TXT",
            "value": "SAMPLE"
        },
        {
            "type": "TXT",
            "value": "RP/PGKGA0101/PGKGA0101"
        },
        {
            "type": "TXT",
            "value": "SAMPLE1"
        }],
        "rhp": {
            "pnrLocator": "K5KRE920171205",
            "primPnrLocator": "K5KRE920171205",
            "tcpNumber": 0,
            "numOfNames": 0
        },
        "rfg": [{
            "dirNum": 1,
            "orgn": "CGK",
            "dstn": "PGK",
            "sumFareCase": null,
            "rqstdCmp": "Y",
            "prefClassCode": "G",
            "rfd": [{
                "rangeNum": 1,
                "startDate": "20180102",
                "endDate": "20180102",
                "startTime": "0000",
                "endTime": "2359",
                "arrDptFlag": 0,
                "rfs": [{
                    "segNum": 1,
                    "crrCode": "GA",
                    "flightNum": "138",
                    "segOrgn": "CGK",
                    "segDstn": "PGK",
                    "cmpCode": "Y",
                    "classCode": "G",
                    "dptDate": "20180102",
                    "dptTime": "1505",
                    "arrTime": "1625",
                    "bkgStatus": "HN"
                }]
            }]
        }]
    }
}
```

### *3.1.2.2 PNR Load*

#### 3.1.2.2.1 Description

The group Materialization process uses selected historical performance to predict the group utilization. It continuously tracks and monitors group behavior until departure; this process is performed by the PNRLOAD.

The group Materialization process uses selected historical performance to predict the group utilization. It continuously tracks and monitors group behavior until departure; this process is performed by the PNRLOAD. The section below describes the detail of the input file format for group tracking.

#### 3.1.2.2.2 Data Fields

There can be multiple PBG records for each input file.

RecordSet: PBG

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| PnrLocator | LOCATOR | Y | | | Y | B32DT20 050812 | PNR Locator |
| PrimPnrLocator | LOCATOR | Y | | | | B32DT20 050812 | Primary PNR Locator |
| RefPnrLocator | LOCATOR | N | | | | B32DT20 050812 | Reference PNR Locator. Don't pass this field if there is no value. |
| TotalRevenue | NUM 10 | N | | | | 100 | The total fare for the entire group to travel on the trip.<br>Don't pass this field if there is no value.<br><br>This value is used to calculate the real fare value. If LCU related fields are populated, then the real fare value will be calculated by LCU related fields instead. Please check the LCU comment under <PMG>. Note: The LCU related fields are LCU, local fare, and exchange rate. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| TotalGroupBooked | NUM 03 | Y | | | | 100 | Total seats booked for the Group.<br>When the whole PNR is cancelled, this value must be 0. Otherwise, this value must not be 0 regardless the non-confirmed status of any PNR itinerary. In other words, this value is not 0 as long as the whole PNR is not cancelled.<br>NOTE: The "Total PNR Group Bookings" can differ from the number of "PNR Segment Group Bookings" (in the PFG record). The difference may be caused, for example by Waitlisted Group's counts, which must also be included in this field. E.G.:<br>If a group of 20 passengers was requested, but only 10 could be confirmed, this field would contain 20, but the fields in the PFG record would have 10 each:<br>ZZ 090 IAH-FCO 10 KK<br>ZZ 090 IAH-FCO 10 WL<br>Another difference could be caused by itineraries with feeders. For example:<br>DFW – IAH – FCO<br>SAT – IAH – FCO<br>In this case, the size of the whole group is of 20 (who will be traveling IAH – FCO). The whole number must be reflected in this field. However, 12 passengers may come from DFW and 8 from SAT. These numbers will have to be reflected in different PFG lines. |
| NumConfirmedNames | NUM 03 | Y | | | | 10 | Total number of individual passenger's Names advised. |
| CancellationIndicatorFlag | FLAG 01 | Y | | | | 0 | Used for complete itinerary cancellation.<br>0 = not canceled (Default)<br>1 = canceled |
| CancellationCode | CHAR 03 | N | | | | | Type of cancellation for future utilization profile. Don't pass this field if there is no value. |
| OutBoundIndicatorFlag | FLAG 01 | Y | | | | 1 | 0 = Inbound<br>1 = Outbound (Default) |
| RqstID | CHAR 10 | N | | | | | Request Identification (ID). Don't pass this field if there is no value. |
| <PMG> | | Y | | | | | Miscellaneous information for the group shell and PNR booking |
| <PFG> | | N | | | | | |

#### *3.1.2.2.2.1 PNR Itinerary*

This record contains Flight Segment level information. There can be multiple PFG records for each PBG record.

All active segments in the itinerary must be captured, including informational segments.

In case of cancellations, two different cases have observed:

- **Total cancellation of the PNR**
  In case of whole group cancellations, some reservations systems will remove all itineraries' information from the PNR. Therefore, Airlines will not be able to supply this record during PNR cancellation. If the cancellation indicator in the PBG record = 1, then no PFG records will be included in the file. However if based on the RES functionality is the information can be provided, it is advisable to send the appropriate data to the PROS Group System.

- **Partial cancellation of the PNR**
  When only a portion (i.e. a segment) of the group has been canceled, the PFG record should be passed with the Airline's canceled status code (e.g.: XX).

RecordSet:PFG

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| ItnLineNum | NUM 02 | Y | | | | 1 | Itinerary Line Number |
| CrrCode | CARRIER | Y | | | | ZZ | Carrier Code |
| FlightNum | FLIGHT 05 | Y | | | | 100 | Flight Number |
| SegOrgn | CITY | Y | | | | FRA | Segment Origin |
| SegDstn | CITY | Y | | | | JFK | Segment Destination |
| DptDate | DATE | Y | | | | 20051031 | Departure Date |
| ClassCode | SYMBOL 02 | Y | | | | H | Class Code |
| BkgStatusCode | CHAR 02 | Y | | | | HK | Booking Status Code |
| DptTime | TIME 04 | Y | | | | 1030 | Departure Time |
| ArrTime | TIME 04 | Y | | | | 1930 | Arrival Time |
| SegGrpBkd | NUM 03 | Y | | | | 10 | Number of Group PNR bookings on the Segment. |
| DateOffset | NUM 02 | Y | | | | 0 | The difference in days between the departure day of traveled segment and the departure day for the first leg of the flight.<br>If the information cannot be provided by the airline, a default value of 99 must be used. In this case, the Group System will retrieve schedule information from the P5 and O&D System to get the Date Offset information. This will cause a slight deterioration in performance. |
| ClassPOS | CHAR 10 | N | | | | | |

### *3.1.2.2.2.2 PNR Information*

There can be one PMG record for each PBG record. IMPORTANT NOTE: For groups that are booked outside of the Group System, this record must be populated.

RecordSet:PMG

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| GroupName | CHAR 50 | Y | | | | 2P1A1/T2P1A120030806 | Group Name |
| ProgramName | CHAR 50 | N | | | | | Program Name. Don't pass this field if there is no value. |
| TourName | CHAR 50 | N | | | | | Tour Name. Don't pass this field if there is no value. |
| CustType | NUM 10 | Y | Customer Type | | | 3 | Customer Type key valie.<br><br>If the information cannot be provided by the airline, a default value of 0 must be used.<br>In this case, the Group System will keep the existing information from Group Shell table for this PNR family. |
| GroupType | CHAR 02 | N | | | | A | Group Type<br>A = ad hoc<br>S = series<br>We strongly receommend that the G6 user can provide correct group type information (A or S). If this value is empty, G6 will keep the old group type value for any existing PNR. G6 will set the default value 'A' if this group type is not provided and this PNR is new to G6. |
| MktOrgn | CITY | N | | | | JFK | Market Origin. Don't pass this field if there is no value. |
| MktDstn | CITY | N | | | | DEL | Market Destination. Don't pass this field if there is no value. |
| CityPOS | CITY | N | | | | JFK | City Point of Sale . Don't pass this field if there is no value. |
| DistPOS | CITY | N | | | | US | District Point of Sale . Don't pass this field if there is no value. |
| AgencyIATA | CHAR 10 | N | Agency | | | A987654321 | Agency IATA. Don't pass this field if there is no value. |
| OfficeCode | CHAR 10 | N | Agency | | | FRALH0860 | Office Code.<br>Either IATA or Office Code must exist. Don't pass this field if there is no value. |
| AgencyName | CHAR 50 | N | Agency | | | ABC TRAVEL AGENCY | Agency Name. Don't pass this field if there is no value. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| LCU | CHAR 03 | N | | | | USD | Local Currency Code. Don't pass this field if there is no exchange rate information. The exchange rate information can be provided in the following filed or in the exchange rate reference table.<br>This field is used to calculate the real fare value for this PNR. Please ensure that the exchange rate information is properly populated before this filed can be used. |
| ExchangeRate | FLT 11.5 | N | | | | 1.5 | Rate-of-Exchange. Don't pass this field if there is no value.<br>G6 will use this value to calculate real fare value if LocalFare and LCU are both provided in the file. Otherwise, G6 will retrieve the exchange rate information from the G6 database for the real fare calculation. |
| SpecialHandlingRequest | CHAR 20 | N | | | | | Special Handle Request. Don't pass this field if there is no value. |
| LocalFare | NUM 10 | N | | | | | Local fare on the PNR level. Don't pass this field if there is no value.<br><br>Please refer to the LCU and ExchangeRate comments above.<br><br>Please check on the 3.1.3.8 on maximum positive number 2147483647 |
| AncillaryRevenue | NUM 10 | N | | | | 150 | This is the sum of the ancillary revenue for the selected options. |
| OutstationID | CHAR 09 | N | | | | OUT11 | Airlines Outstation/Sales office where request originates from. |

#### 3.1.2.2.3 Processing Rule

None.

#### 3.1.2.2.4 Loader Behavior

Insert/Update

#### 3.1.2.2.5 Schema

The content of schema can be found at /config/interface/xsd/pnrload.xsd file.

#### 3.1.2.2.6 Example

```
<?xml version="1.0"?>
<PnrLoad>
  <PnrData LoadingDate="20050806">
   <PBG>
```

```
<PnrLocator>2P1A1/T2P1A120030806</PnrLocator>
<PrimPnrLocator>2P1A1/T2P1A120030806</PrimPnrLocator>
<RefPnrLocator></RefPnrLocator>
<TotalRevenue>0</TotalRevenue>
<TotalGroupBooked>100</TotalGroupBooked>
<NumConfirmedNames>10</NumConfirmedNames>
<CancellationIndicatorFlag>0</CancellationIndicatorFlag>
<CancellationCode></CancellationCode>
<OutBoundIndicatorFlag>1</OutBoundIndicatorFlag>
<RqstID></RqstID>
<PMG>
   <GroupName>GROUP</GroupName>
   <ProgramName>Student Program </ProgramName>
   <TourName>Major city tour</TourName>
   <CustType>0</CustType>
   <GroupType>A</GroupType>
   <MktOrgn>FRA</MktOrgn>
   <MktDstn>JFK</MktDstn>
   <CityPOS>VER</CityPOS>
   <Request>EU</Request>
   <AgencyIATA>IATA</AgencyIATA>
   <AgencyName>Travel agency</AgencyName>
   <LCU>EU</LCU>
   <ExchangeRate>0</ExchangeRate>
  <SpecialHandlingRequest>special handling</SpecialHandlingRequest>
  <LocalFare>0</LocalFare>
</PMG>
<PFG>
   <ItnLineNum>1</ItnLineNum>
   <CrrCode>ZZ</CrrCode>
   <FlightNum>400</FlightNum>
   <SegOrgn>FRA</SegOrgn>
   <SegDstn>JFK</SegDstn>
   <DptDate>20040621</DptDate>
   <ClassCode>H</ClassCode>
   <BkgStatusCode>HK</BkgStatusCode>
   <DptTime>1030</DptTime>
   <ArrTime>1305</ArrTime>
   <SegGrpBkd>100</SegGrpBkd>
   <DateOffset>0</DateOffset>
   <ClassPOS>ALL</ClassPOS>
</PFG>
</PBG>
<PBG>
<PnrLocator>2P1A2/T2P1A220030806</PnrLocator>
<PrimPnrLocator>2P1A1/T2P1A120030806</PrimPnrLocator>
<RefPnrLocator>2P1A1/T2P1A120030806</RefPnrLocator>
<TotalRevenue>0</TotalRevenue>
<TotalGroupBooked>40</TotalGroupBooked>
<NumConfirmedNames>4</NumConfirmedNames>
<CancellationIndicatorFlag>0</CancellationIndicatorFlag>
<CancellationCode>CNL</CancellationCode>
<OutBoundIndicatorFlag>1</OutBoundIndicatorFlag>
<RqstID>A1234567</RqstID>
<PMG>
   <GroupName>CHILD</GroupName>
   <ProgramName>Child program</ProgramName>
```

```
      <TourName>Child tour</TourName>
      <CustType>0</CustType>
      <GroupType>A</GroupType>
      <MktOrgn>FRA</MktOrgn>
      <MktDstn>JFK</MktDstn>
      <CityPOS>VER</CityPOS>
      <Request>EU</Request>
      <AgencyIATA>IATA</AgencyIATA>
      <OfficeCode>OfficeCode</OfficeCode>
      <AgencyName>World Travel</AgencyName>
      <LCU>EU</LCU>
      <ExchangeRate>0</ExchangeRate>
     <SpecialHandlingRequest>Special handling</SpecialHandlingRequest>
     <LocalFare>0</LocalFare>
   </PMG>
   <PFG>
      <ItnLineNum>1</ItnLineNum>
      <CrrCode>ZZ</CrrCode>
      <FlightNum>150</FlightNum>
      <SegOrgn>FRA</SegOrgn>
      <SegDstn>JFK</SegDstn>
      <DptDate>20040621</DptDate>
      <ClassCode>H</ClassCode>
      <BkgStatusCode>HK</BkgStatusCode>
      <DptTime>1030</DptTime>
      <ArrTime>1305</ArrTime>
      <SegGrpBkd>100</SegGrpBkd>
      <DateOffset>0</DateOffset>
      <ClassPOS>All</ClassPOS>
   </PFG>
   </PBG>
 </PnrData>
 </PnrLoad>
```

### *3.1.2.3 Acceptance Load*

#### 3.1.2.3.1 Description

The Acceptance Queue contains the options accepted by the Outstation. The accepted options will be compared with the time/date stamp on the held record. If the record is not stale (based on a system parameter), the options accepted by the outstation will be automatically processed. However, if anything in the acceptance file has changed, it will not be eligible for auto-acceptance and further evaluation or manual review is required. AAP (Autopilot for Acceptance Queue) will process the Acceptance Requests automatically if the AAP flag is set during the Negotiation stage.

Note that the negotiation process does not allow multiple iterations of this exchange. There is no facility that to send back to the Outstation a request that has gone back to the Acceptance Queue. There is no facility to get back to the holding queue from the acceptance queue. If a request is stale, or for any other reason the Group user does not believe the group should be accepted (e.g.: the fare is unacceptable), the request should be treated as a new one.

### 3.1.2.3.2 Data Fields

#### *3.1.2.3.2.1 Header information*

This record contains general request level information. There will be one AHG record for each request.

DatasetName: AcceptanceLoad
RecordSet: AHG (for XML) and ahg (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| RqstID (XML)<br>rqstID (JSON) | CHAR 10 | Y | | | | Group433 | Unique Reference Number used for identifying a Group Request application. |
| OutstationID (XML)<br>outstationID (JSON) | CHAR 09 | Y | | | | OUT11 | Airlines Outstation/Sales office where request originates from. |
| RequestDate (XML)<br>requestDate (JSON) | DATE | Y | | | | 20050930 | Date group booking request submitted to G6.<br>YYYYMMDD |
| RequestTime (XML)<br>requestTime (JSON) | TIME 04 | Y | | | | 1130 | Time group booking request submitted to G6.<br>Format: 0000 – 2359 |
| AcceptedFare (XML)<br><br>acceptedFare (JSON) | NUM 10 | N | | | | 500 | Accepted fare. This fare is NOT required on the request header level if they are provided in the direction level below (AFG). In the case where the fares are provided on both levels (header and direction) GRMS will use the fares in the direction level with higher priority and override the fare in the request header level with the sum of all the directions.<br><br>Please check on the 3.1.3.8 on maximum positive number 2147483647.<br><br>Final note: The Accepted fare must be provided in either the request header level or direction level. If neither one is provided then it GRMS considered as an error. |
| LocalFare (XML)<br><br>localFare (JSON) | NUM 10 | N | | | | 500 | The Local Fare value in the Local Currency Code for the acceptance fare. If this is not provided GRMS will calculate it based on the exchange rate used in the original request.<br>Do not pass this field if the value is NULL.<br>Please check on the 3.1.3.8 on maximum positive number 2147483647. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| StaleFlag (XML) staleFlag (JSON) | FLAG 01 | Y | | | | 0 | This field is still under design. Current program will ignore this field. |
| <AFG> (XML) "afg" (JSON) | | Y | | | | | Request Direction/Option information. Normal option contains one option per direction. Break-up options will have multiple options per direction. |
| <Note> (XML) "note" (JSON) | | Y | | | | | Note for the acceptance. However, duplicate Note entries can occur if the same Notes sent 72tilizat in the Holding file/Request files are returned with the Acceptance file. This is because the acceptance process expects only new notes will be sent in the acceptance file. |

#### *3.1.2.3.2.2 Direction information*

This AFG record contains directional information for the approved options.

RecordSet: AFG (for XML) and afg (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DirNum (XML) dirNum (JSON) | NUM 02 | Y | | | | 1 | Direction number |
| AcceptedFare (XML) acceptedFare (JSON) | NUM 10 | N | | | | 200 | Accepted fare. This fare is NOT required on the direction level if one has already been provided in the request header level (AHG). In the case where the fares are provided on both levels (header and direction) GRMS will use the fares in the direction level with higher priority and override the fare in the request header level with the sum of all the directions. If the fare is provided in at least 1 direction then all direction(s) must have the fare provided, otherwise GRMS considered this as error. Final note: The Accepted fare must be provided in either the request header level or direction level. If neither one is provided then it GRMS considered as an error. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| LocalFare (XML)<br><br>localFare (JSON) | NUM 10 | N | | | | | The Local Fare value in the Local Currency Code for the acceptance fare. If this is not provided GRMS will calculate it based on the exchange rate used in the original request.<br>Do not pass this field if the value is NULL.<br>Please check on the 3.1.3.8 on maximum positive number 2147483647. |
| <AFO> (XML)<br><br>"afo" (JSON) | | Y | | | | | Option information |

#### *3.1.2.3.2.3 Option information*

This AFO record contains information for the approved options per AFG record.

RecordSet: AFO (for XML) and afo (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| OptID (XML)<br><br>optID (JSON) | NUM 04 | Y | | | | 1 | Option ID. |
| SubOptID (XML)<br><br>subOptID (JSON) | NUM 04 | Y | | | | 1 | Sub Option ID. When Sub Option is not the same with Option ID, it indicates option break up. |
| ClassCode (XML)<br><br>classCode (JSON) | SYMBOL 02 | Y | | | | M | Class symbol for this option |
| ClassPOS (XML)<br><br>classPOS (JSON) | CHAR 10 | N | | | | M | |
| <AFF> (XML)<br><br>"aff" (JSON) | | N | | | | | |

#### 3.1.2.3.2.3.1 Option Flight information

This AFF record contains information for the option flight per AFF record.

There can be multiple AFF records. Each record contains the information for each traveled flight constituting each itinerary option.

RecordSet: AFF (for XML) and aff (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| CrrCode (XML)<br><br>crrCode (JSON) | CARRIER | Y | | | | ZZ | Carrier Code |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| FlightNum (XML)<br><br>flightNum (JSON) | FLIGHT 05 | Y | | | | 441 | Flight number |
| ClassCode (XML)<br><br>classCode (JSON) | SYMBOL 02 | Y | | | | M | Class symbol for this flight |
| ClassPOS (XML)<br><br>classPOS (JSON) | CHAR 10 | N | | | | ALL | Class Point of Sale |

#### *3.1.2.3.2.4 Note information*

There can be multiple Note records for each request.

RecordSet: Note

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Type (XML)<br><br>type (JSON) | CHAR 04 | Y | | | Y | CTC | The identifier for note text. There is no restriction of unique note type. |
| Value (XML)<br><br>value (JSON) | CHAR 256 | Y | | | | | The text content |

### 3.1.2.3.3 Processing Rules

None

### 3.1.2.3.4 Loader Behavior

Insert/Update

### 3.1.2.3.5 Schema

The content of schema can be found at /config/interface/xsd/acc.xsd file.

### 3.1.2.3.6 Example

```
<?xml version="1.0" ?>
<AcceptanceLoad>
  <AHG>
    <RqstID>Group433</RqstID>
    <OutstationID>OUT11</OutstationID>
    <RequestDate>20050930</RequestDate>
    <RequestTime>1130</RequestTime>
    <AcceptedFare>500</AcceptedFare>
    <LocalFare>500</LocalFare>
    <StaleFlag>0</StaleFlag>
    <Note>
       <Type>CTC</Type>
       <Value> Boris Walden</Value>
    </Note>
    <Note>
```

```
      <Type>RCV</Type>
      <Value> Jurgen Singer </Value>
   </Note>
   <Note>
      <Type>SSR</Type>
      <Value> SSR value </Value>
   </Note>
   <Note>
      <Type>OSI</Type>
      <Value> OSI value</Value>
   </Note>
   <Note>
      <Type>IMP</Type>
      <Value>IMP value</Value>
   </Note>
   <Note>
      <Type>HLD</Type>
      <Value>HLD value</Value>
   </Note>
   <Note>
     <Type>STO</Type>
  <Value>This is a value to be stored in the Group Shell after PNRGen</Value>
   </Note>
   <Note>
      <Type>TXT</Type>
      <Value>This is a PNR Remark</Value>
   </Note>
   <AFG>
      <DirNum>1</DirNum>
      <AcceptedFare>500</AcceptedFare>
      <LocalFare>500</LocalFare>
      <AFO>
         <OptID>1</OptID>
         <SubOptID>1</SubOptID>
         <ClassCode>Y</ClassCode>
         <ClassPOS>USA</ClassPOS>
     </AFO>
     </AFG>
  </AHG>
</AcceptanceLoad>
```

### 3.1.2.3.7 JSON load via Acceptance rest loader

Rest Endpoint: **/group/rest/submitAcceptRequest**
Example: http://localhost:8080/group/rest/submitAcceptRequest

### 3.1.2.3.8 Example JSON content for acceptance rest loader

```
{
   "ahg":{
      "rqstID":"A1078",
      "outstationID":"HOMEOFFIC",
      "requestDate":"20180613",
      "requestTime":"1345",
      "acceptedFare":0.2,
      "localFare":2.1,
      "staleFlag":0,
      "afg":[
         {
            "dirNum":1,
            "acceptedFare":1.0,
```

```
            "localFare":1.0,
            "afo":{
               "optID":1,
               "subOptID":1.256,
               "classCode":"S",
               "aff":[
               {
                 "crrCode": "QR",
                 "flightNum": "123",
                 "classCode": "G",
                 "classPOS": "H"
               },
               {
                 "crrCode": "QR",
                 "flightNum": "124",
                 "classCode": "G",
                 "classPOS": "H"
               }
               ]
            }
         },
         {
            "dirNum":2,
            "acceptedFare":1.0,
            "localFare":1.0,
            "afo":{
               "optID":1,
               "subOptID":1,
               "classCode":"S"
            }
         }
      ],
      "note":[
         {
            "type":"CTC",
            "value":"Accepted"
         }
      ]
   }
}
```

### *3.1.2.4 Series Request Load*

#### 3.1.2.4.1 Description

The Series loader is a non-interactive program to load the Series requests into the database. The Series requests are in a XML file given to the loader. The loading program also will calculate the utilization for each request and store the results in the database. New data will be stored automatically in the corresponding tables.

The PROS Group System will accept whatever data is passed to it. If the request file has the necessary data (correct or incorrect) and is formatted correctly, it will be loaded. In the case that the corruption was to the formatting or so severe that the request is not loadable, it will not be loaded. It is the Airlines' responsibility to create the clean load file and put it in the data directory.

#### 3.1.2.4.2 Data Fields

##### *3.1.2.4.2.1 Request Header*

This record contains general request level information. There will be one SHG record for each request.

RecordSet: SHG (for XML) and shg (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| RqstID (XML)<br><br>RqstID (JSON) | CHAR 10 | Y | | | Y | Group433 | Unique Reference Number used for identifying a Group Request application. |
| OutstationID (XML)<br><br>outstationID (JSON) | CHAR 09 | Y | | | Y | OUT11 | Airlines Outstation/Sales office where request originates from. |
| RequestDate (XML)<br><br>requestDate (JSON) | DATE | Y | | | Y | 20050930 | Date group booking request submitted to G6.<br>Format: YYYYMMDD |
| RequestTime (XML)<br><br>requestTime (JSON) | TIME 04 | Y | | | Y | 1130 | Time group booking request submitted to G6.<br>Format: 0000 – 2359 |
| GroupName (XML)<br><br>groupName (JSON) | CHAR 50 | Y | | | | U OF H GROUP | Name of the group for use in PNR Creation. |
| ProgramName (XML)<br>programName (JSON) | CHAR 50 | N | | | | Summer Travel | Additional identification for Group. |
| TourName (XML)<br><br>tourName (JSON) | CHAR 50 | N | | | | Excursion | Additional identification for Group.<br>Don't pass this field if there is no value. |
| CustomerType (XML)<br><br>customerType (JSON) | NUM 10 | Y | Customer type | | | 3 | Number used to identify category type (e.g.: Leisure, Cruise, Sports, etc.) |
| AgencyIata (XML)<br><br>agencyIata (JSON) | CHAR 10 | N | Agency | | | A987654321 | GDS or other Airlines IATA reference.<br>Don't pass this field if there is no value. |
| OfficeCode (XML)<br><br>officeCode (JSON) | CHAR 10 | N | Agency | | | FRALH0860 | Office Code.<br>Either IATA or Office Code must exist.<br>Don't pass this field if there is no value. |
| AgencyName (XML)<br><br>agencyName (JSON) | CHAR 50 | N | Agency | | | Discount Travelers | GDS or other Airlines Name.<br>Don't pass this field if there is no value. |
| GroupType (XML)<br><br>groupType (JSON) | CHAR 02 | Y | | | | S | A = Ad hoc<br>S = Series |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| WorkQueueID (XML)<br><br>workQueueID (JSON) | NUM 03 | N | | | | 0 | Numerical ID of Work Queue request will be assigned to.<br>0= Central (Default).<br>Don't pass this field if there is no value. |
| PartialConfirmFlag (XML)<br><br>partialConfirmFlag (JSON) | FLAG 01 | Y | | | | 1 | 2 = Allow Partial Confirmation<br>0 = Do not allow Partial Confirmation (default)<br>This flag is used for the outside reservation window feature. |
| PreferredFlag (XML)<br><br>preferredFlag (JSON) | FLAG 01 | Y | | | | 1 | Flag to indicate that CCITIN must return from RES options with requested flight only.<br><br>1 = Yes (Default)<br>0 = No |
| SeriesID (XML)<br><br>seriesID (JSON) | NUM 38 | N | | | | | This field is designed specifically for the Kebut interface.<br>Normal series request should not provide this information. |
| AgentURL (XML)<br>agentURL (JSON) | CHAR128 | N | | | | | The URL to send responses back to the agent. |
| <SPG> (XML)<br><br>"spg" (JSON) | | | | | | | Pattern information |

#### *3.1.2.4.2.2 Pattern Information*

This record contains pattern level information. There can be multiple SPG records for each request – one for each pattern.

RecordSet: SPG (for XML) and spg (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| PattNum (XML)<br><br>pattNum (JSON) | NUM 02 | Y | | | Y | 1 | Number of pattern block |
| CityPOS (XML)<br><br>cityPOS (JSON) | CITY | Y | | | | IAH | Departure Airport for the City where the request originated.<br>In case of new Agency IATA, this field will populate the City POS field in the Agency table in G6. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DistPOS (XML)<br><br>distPOS (JSON) | CITY | Y | | | | US | Country or Airport code of the City where request originated.<br>In case of new Agency IATA, this field will populate the District POS field in the Agency table in G6. |
| MarketOrg (XML)<br><br>marketOrg (JSON) | CITY | Y | Market O&D | | | IAH | Default to first direction of origin. |
| MarketDstn (XML)<br><br>marketDstn (JSON) | CITY | Y | Market O&D | | | FRA | Default to first direction of destination. |
| CommPnct (XML)<br><br>commPnct (JSON) | NUM 02 | Y | | | | 0 | Agent commission percentage.<br>Must be a whole number. |
| LCU (XML)<br><br>Lcu (JSON) | CHAR 03 | N | | | | USD | Currency code of the Airlines Outstation or RES Office (i.e. LAX office = USD, LON office = GBP).<br>Don't pass this field if there is no value. |
| DealIndicatorFlag (XML)<br><br>dealIndicatorFlag (JSON) | CHAR 02 | N | | | | G | This field will indicate the Deal Type for each request (Request Header Level). It can have 3 values – G (Ad Hoc), S (Series), M (Ad Hoc). The M would indicate that the Deal will be based on a agreed upon Market Fare |
| TripType (XML)<br><br>tripType (JSON) | CHAR 02 | N | | | | RT | This field would indicate if the Group Request is for Round Trip (RT) or One Way (OW). Open Jaw Group Request will be send to G6 as a OJ. |
| DiscountPercent (XML)<br><br>discountPercent (JSON) | CHAR 03 | N | | | | 15 | An extra discount percentage for this request associated with the City POS |
| <Note> (XML)<br><br>Note (JSON) | | N | | | | | Note |
| <SFI> (XML)<br><br>Sfi (JSON) | | Y | | | | | Fare information |
| <SDT> (XML)<br><br>Sdt (JSON) | | Y | | | | | Departure Information |

#### *3.1.2.4.2.3 Fare Information*

This record contains fare information for the pattern. There can be multiple SFI records for each SPG record. Note that the Fare periods must be contiguous and cover the entire range requested.

RecordSet: SFI (for XML) and sfi (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| EffDate (XML)<br><br>effDate (JSON) | DATE | Y | | | Y | 20060101 | This field contains the date starting from which the Quoted Fare is valid.<br>Format: YYYYMMDD |
| DisContDate (XML)<br><br>disContDate (JSON) | DATE | Y | | | Y | 200600331 | This field contains the last departure date the Quoted Fare is valid.<br>Format: YYYYMMDD |
| QuotedFare (XML)<br><br>quotedFare (JSON) | NUM 10 | N | | | | 500 | This fare is for the individual trip. In Series, it is for each departure.<br>This filed is changed to optional for currency conversion. Either quoted fare or local fare must be provided in SFI.<br>Please check on the 3.1.3.8 on maximum positive number 2147483647. |
| LocalFare (XML)<br><br>localFare (JSON) | NUM 10 | N | | | | 500 | Fare being charged per person in Local Currency Code as defined above.<br>If this field is provided, the exchange rate information must be available for the quoted fare conversion.<br>Please check on the 3.1.3.8 on maximum positive number 2147483647. |
| SeasonIndicator (XML)<br><br>seasonIndicator (JSON) | CHAR 05 | N | | | | | Identification code meant to define a particular season associated to the defined Fare range. |

#### *3.1.2.4.2.4 Departure Information*

This record contains departure/seat information for the pattern. There must be only one SDT record for each complete single group itinerary.

RecordSet: SDT (for XML) and sdt (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DptNum (XML)<br><br>dptNum (JSON) | NUM 02 | Y | | | Y | 1 | Departure number starting from1. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| SeatsRqstd (XML)<br><br>seatsRqstd (JSON) | NUM 03 | Y | | | | 20 | |
| SeatsPasses (XML)<br><br>seatsPasses (JSON) | NUM 03 | Y | | | | 0 | Non-revenue tickets. |
| ProcessId (XML)<br><br>processId (JSON) | NUM 10 | N | | | | 1234567 | This filed is designed for the special communication to the client. Currently, it will pass back through the confirmation file. |
| PnrLocator (XML)<br><br>pnrLocator (JSON) | LOCATOR | N | | | | | This filed is designed specifically for the Kebut interface.<br>Normal series request must not provide this information. |
| <SFG> (XML)<br><br>"sfg" (JSON) | | Y | | | | | Direction Information |

#### *3.1.2.4.2.5 Direction Information*

This record contains departure/flight/itinerary information for the pattern.

RecordSet: SFG (for XML) and sfg (JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DirNum (XML)<br><br>dirNum (JSON) | NUM 02 | Y | | | Y | 1 | Direction number starting from 1. The outbound direction must be direction 1. |
| Orgn (XML)<br><br>orgn (JSON) | CITY | Y | | | | IAH | Boarding point. |
| Dstn (XML)<br><br>dstn (JSON) | CITY | Y | | | | FRA | Off point. |
| SumFareCase (XML)<br><br>sumFareCase (JSON) | CHAR02 | N | | | | CW | Sum of marginal fares for each segment.<br>If blank value or not providing the element, use Default set up in SysParms. |
| DenyBoardingFactor (XML)<br><br>denyBoardingFactor (JSON) | CHAR 02 | N | | | | | Factor used to load marginal fare according to Fight/date demand levels.<br>If blank value or not providing the element, use Default set up in SysParms. |
| RqstdCmp (XML)<br><br>rqstdCmp (JSON) | SYMBOL 02 | Y | Compartment Cabin mapping | | | M | Compartment code identifying requested cabin. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| PrefClassCode (XML)<br><br>prefClassCode (JSON) | SYMBOL 02 | N | | | | M | Preferred directional booking Class Code.<br><br>Don't pass this field if there is no value. |
| PrefClassPOSCode (XML)<br><br>prefClassPOSCode (JSON) | CHAR 10 | N | | | | | Preferred directional booking Class POS.<br><br>Don't pass this field if there is no value. |
| DptDate (XML)<br><br>dptDate (JSON) | DATE | Y | | | | 20050831 | Start Date for preferred travel.<br>Format: YYYYMMDD |
| StartTime (XML)<br><br>startTime (JSON) | TIME 04 | Y | | | | 0000 | Start Time for preferred travel.<br>Format: 0000 – 2359 |
| EndTime (XML)<br><br>endTime (JSON) | TIME 04 | Y | | | | 2359 | End Time for preferred travel.<br>Format: 0000 – 2359 |
| ArrDptFlag (XML)<br><br>arrDptFlag (JSON) | FLAG 01 | Y | | | | 0 | Indicates if Dates/Times advised above relate to Departure or Arrival.<br>1 = Arrival<br>0 = Departure (Default) |
| StopOverIndicator (XML)<br><br>stopOverIndicator (JSON) | NUM 01 | N | | | | 1 | Each direction will have a Stop Over Indicator Flag. If the Flag is 1, then the Quoted Fare includes a Stop Over Cost for that direction. So G6 would need to take that into consideration during the calculation of Marginal Fares. Note that here the stop over means that the time between the 2 connecting flight is between 4 and 24 hours and the Group would like to stay in a hotel for that duration. (Do not confuse this with the way we are used to defining stop over – which usually means a new direction !). |
| StopOverCost (XML)<br><br>stopOverCost (JSON) | NUM 10 | N | | | | | This field will denote the Stop Over cost for that particular direction |
| <SFP> (XML)<br><br>"sfp" (JSON) | | N* | | | | | Request Segment information<br><br>This segment information is required if the ORW setting (out_of_reservation_window) in pros.dtd is true. |

#### *3.1.2.4.2.6 Flight Information*

This record contains Specific flight information for the pattern.

RecordSet: SFP (for XML) and sfp (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| SegNum (XML)<br>segNum (JSON) | NUM 02 | Y | | | Y | 1 | The range number within each direction will start from 1. |
| CrrCode (XML)<br>crrCode (JSON) | CARRIER | Y | | | | ZZ | Carrier Code |
| FlightNum (XML)<br>flightNum (JSON) | FLIGHT 05 | Y | | | | 441 | Flight number |
| DptDate (XML)<br>dptDate (JSON) | DATE | Y | | | | 20050831 | Format: YYYYMMDD. Don't pass this field if there is no value. |
| SegOrgn (XML)<br>segOrgn (JSON) | CITY | N | | | | IAH | Travel Segment Origin. Group boarding point.. Don't pass this field if there is no value. |
| SegDstn (XML)<br>segDstn (JSON) | CITY | N | | | | FRA | Travel Segment Destination. Group off-boarding point. Don't pass this field if there is no value. |
| ClassCode (XML)<br>classCode (JSON) | SYMBOL 02 | N | | | | M | Class Code. Currently, this filed is reserved for the Kebut process. Don't pass this field if there is no value. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| SetNum (XML)<br><br>setNum (JSON) | NUM 02 | N | | | | 1 | This set number is used mainly for the ORW (booking outside of res window) module.<br>By default, the set number will be 1. The set number will start from 1. This set number is used to group travel flights into a logical set. Different travel options can have different set of preferred flights.<br><br>For example, flight 100 HOU-TPE as set 1. Flight 101 HOU-TPE as set 2. This example shows different preferred (set) options.<br>Another example shows logical group of flights. Flight 102 HOU-TYO and flight 103 TYO-TPE will both be set 1. Flight 103 TYO-TPE will be the connecting flight for HOU-TPE trip.<br><br>Although G6 can support preferred flight of many sets, we recommend that this set number can be limited to 3. More distinct set numbers in the loader file may subsequently increase the processing time. |

#### *3.1.2.4.2.7 Note Information*

There can be multiple Note records for each pattern block (i.e. for each SPG record). The Note record can be used to pass information such as OSI, SSR, etc.

RecordSet: Note (for XML) and note (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Type (XML)<br><br>type (JSON) | CHAR 04 | Y | | | Y | CTC | The identifier for note text. There is no restriction of unique note type. |
| Value (XML)<br><br>value (JSON) | CHAR 256 | Y | | | | | The text content |

### 3.1.2.4.3 Process Rules

None

### 3.1.2.4.4 Loader Behavior

Insert/Update

#### 3.1.2.4.5 Schema

The content of schema can be found at /config/interface/xsd/series.xsd file.

#### 3.1.2.4.6 Examples

```
<?xml version=”1.0” ?>
<SeriesLoad>
  <SHG>
    <RqstID>Group433</RqstID>
    <OutstationID>OUT11</OutstationID>
    <RequestDate>20050930</RequestDate>
    <RequestTime>1130</RequestTime>
    <GroupName>U OF H GROUP</GroupName>
    <ProgramName>Summer Travel</ProgramName>
    <TourName>Excursion</TourName>
    <CustomerType>3</CustomerType>
    <AgencyIata>A987654321</AgencyIata>
    <OfficeCode>FRALH0860</OfficeCode>
    <AgencyName>Discount Travellers</AgencyName>
    <GroupType>S</GroupType>
    <PartialConfirmFlag>0</PartialConfirmFlag>
    <PreferredFlag>1</PreferredFlag>
    <SPG>
        <PattNum>1</PattNum>
        <CityPOS>JFK</CityPOS>
        <DistPOS>US</DistPOS>
        <MarketOrg>JFK</MarketOrg>
        <MarketDstn>DEL</MarketDstn>
        <CommPnct>0</CommPnct>
        <LCU>USD</LCU>
        <Note>
            <Type>CTC</Type>
            <Value> Boris Walden</Value>
        </Note>
        <Note>
            <Type>RCV</Type>
            <Value> Jurgen Singer </Value>
        </Note>
        <Note>
            <Type>SSR</Type>
            <Value> SSR value </Value>
        </Note>
        <Note>
            <Type>OSI</Type>
            <Value> OSI value</Value>
        </Note>
        <Note>
            <Type>IMP</Type>
            <Value>IMP value</Value>
        </Note>
        <Note>
            <Type>HLD</Type>
            <Value>HLD value</Value>
        </Note>
        <Note>
      <Type>STO</Type>
      <Value>This is a value to be stored in the Group Shell</Value>
        </Note>
        <Note>
            <Type>TXT</Type>
            <Value>This is a PNR Remark</Value>
        </Note>
        <SFI>
            <EffDate>20060201</EffDate>
```

```
            <DisContDate>20060215</DisContDate>
            <QuotedFare>600</QuotedFare>
            <LocalFare>600</LocalFare>
        </SFI>
        <SFI>
            <EffDate>20060216</EffDate>
            <DisContDate>20060228</DisContDate>
            <QuotedFare>640</QuotedFare>
            <LocalFare>640</LocalFare>
        </SFI>
        <SDT>
            <DptNum>1</DptNum>
            <SeatsRqstd>20</SeatsRqstd>
            <SeatsPasses>1</SeatsPasses>
            <SFG>
                <DirNum>1</DirNum>
                <Orgn>JFK</Orgn>
                <Dstn>DEL</Dstn>
                <SumFareCase></SumFareCase>
                <DenyBoardingFactor></DenyBoardingFactor>
                <RqstdCmp>M</RqstdCmp>
                <PrefClassCode></PrefClassCode>
                <PrefClassPOSCode></PrefClassPOSCode>
                <DptDate>20060203</DptDate>
                <StartTime>0800</StartTime>
                <EndTime>0600</EndTime>
                <ArrDptFlag>0</ArrDptFlag>
           <SFP>
              <SegNum>1</SegNum>
              <CrrCode>ZZ</CrrCode>
              <FlightNum>00133</FlightNum>
              <DptDate>20060203</DptDate>
             <SetNum>1</SetNum>

           </SFP>
            <SFP>
                <SegNum>2</SegNum>
                <CrrCode>ZZ</CrrCode>
                <FlightNum>00190</FlightNum>
                <DptDate>20060203</DptDate>
                <SetNum>2</SetNum>    // if SetNum is 1, then flight 190 is the
                    connecting flight of flight 133. If SetNum is 2, then flight is on
                    the same market as flight 133. SetNum 2 represents a different
                    option of the same market.

            </SFP>
      </SFG>
      </SDT>
  </SPG>
  </SHG>
</SeriesLoad>
```

### 3.1.2.4.7 Series JSON load via series request loader

End point: **/group/rest/submitSeriesRequest**
Example: http://localhost:8080/group/rest/submitSeriesRequest

### 3.1.2.4.8 Example Series JSON content for series rest loader

```
{
    "shg": {
```

```
"rqstID": "STEST9",
"outstationID": "homeoffic",
"requestDate": "20180606",
"requestTime": "1737",
"groupName": "ABC",
"customerType": 1,
"agencyIata": "8627532",
"agencyName": ": AL MASAOOD TRAVEL",
"groupType": "S",
"partialConfirmFlag": 0,
"preferredFlag": 0,
"agentURL": "http://localhost:8090/groupagent/rest/",
"spg": [{
    "pattNum": 1,
    "cityPOS": "AUH",
    "distPOS": "AE",
    "marketOrg": "COK",
    "marketDstn": "DOH",
    "commPnct": 0,
    "lcu": "ABC",
    "tripType": "OW",
    "note": [{
    "type": "TXT",
    "value": "SAMPLE"
    },
    {
    "type": "TXT",
    "value": "RP/PGKGA0101/PGKGA0101"
    },
    {
    "type": "TXT",
    "value": "SAMPLE1"
    }],
    "sfi": [{
        "effDate": "20180531",
        "disContDate": "20180531",
        "quotedFare": 3.0,
        "localFare": 2.0
    }],
    "sdt": [{
        "dptNum": 1,
        "seatsRqstd": 10,
        "seatsPasses": 0,
        "sfg": [{
            "dirNum": 1,
            "orgn": "COK",
            "dstn": "DOH",
            "rqstdCmp": "Y",
            "dptDate": "20180531",
            "startTime": "0000",
            "endTime": "2359",
            "arrDptFlag": 0,
            "sfp" [{
                "segNum": 1,
                "crrCode": "ZZ",
                "flightNum": "123",
                "dptDate": "20181231"
            }]
        }]
    }]
},
{
    "pattNum": 2,
```

```
"cityPOS": "AUH",
"distPOS": "AE",
"marketOrg": "COH",
"marketDstn": "DOK",
"commPnct": 0,
"lcu": "USD",
"tripType": "RT",
"sfi": [{
    "effDate": "20181231",
    "disContDate": "20190131",
    "quotedFare": 1.79,
    "localFare": 2.0
}],
"sdt": [{
    "dptNum": 1,
    "seatsRqstd": 10,
    "seatsPasses": 0,
    "sfg": [{
        "dirNum": 1,
        "orgn": "COH",
        "dstn": "DOK",
        "rqstdCmp": "Y",
        "dptDate": "20181231",
        "startTime": "0000",
        "endTime": "2359",
        "arrDptFlag": 0,
        "sfp": [{
            "segNum": 1,
            "crrCode": "QR",
            "flightNum": "00515",
            "dptDate": "20181231",
            "segOrgn": "COK",
            "segDstn": "DOH"
        }]
    },
    {
        "dirNum": 2,
        "orgn": "DOK",
        "dstn": "COH",
        "rqstdCmp": "Y",
        "dptDate": "20190131",
        "startTime": "0000",
        "endTime": "2359",
        "arrDptFlag": 0,
        "sfp": [{
            "segNum": 1,
            "crrCode": "QR",
            "flightNum": "00516",
            "dptDate": "20190131",
            "segOrgn": "DOK",
            "segDstn": "COH"
        }]
    }]
},
{
    "dptNum": 2,
    "seatsRqstd": 10,
    "seatsPasses": 0,
    "sfg": [{
        "dirNum": 1,
        "orgn": "COH",
        "dstn": "DOK",
        "rqstdCmp": "Y",
```

```
                    "dptDate": "20190128",
                    "startTime": "0000",
                    "endTime": "2359",
                    "arrDptFlag": 0,
                    "sfp": [{
                        "segNum": 1,
                        "crrCode": "QR",
                        "flightNum": "00515",
                        "dptDate": "20190128",
                        "segOrgn": "COK",
                        "segDstn": "DOH"
                    }]
                },
                {
                    "dirNum": 2,
                    "orgn": "DOK",
                    "dstn": "COH",
                    "rqstdCmp": "Y",
                    "dptDate": "20190228",
                    "startTime": "0000",
                    "endTime": "2359",
                    "arrDptFlag": 0,
                    "sfp": [{
                        "segNum": 1,
                        "crrCode": "QR",
                        "flightNum": "00516",
                        "dptDate": "20190228",
                        "segOrgn": "DOK",
                        "segDstn": "COH"
                    }]
                }]
            }]
        }]
    }
}
```


### *3.1.2.5 Reject Request Load*

#### 3.1.2.5.1 Description

The Reject loader is a non-interactive program to load the Reject requests into the database. The Reject requests are in a XML file given to the loader. The loading program also will delete this request in the request queue. In addition, the loading program will update the request report with the reason for rejection and set the rejected flag. This Reject loader is only used for the Group Agent user to reject the request during the negotiation process. However, the same loder can be used as a general purpose tool to reject a non-confirmed request. It is the Group user's responsibility to make sure that the request stage of the request is appropriate for rejection. For example, a series request with PNRGen failed stage is not appropriate for the rejection. It is due to the reason that some departures may have successfully finished the PNR generation process even some other departures may have failed.

The PROS Group System will accept whatever data is passed to it. If the request file has the necessary data (correct or incorrect) and is formatted correctly, it will be loaded. In the case that the corruption was to the formatting or so severe that the request is not loadable, it will not be loaded. It is the Airlines' responsibility to create the clean load file and put it in the data directory.

#### 3.1.2.5.2 Data Fields

##### *3.1.2.5.2.1 Request Header*

This record contains general request level information. There will be one RJG record for each request.

RecordSet: RJG (for XML) and rjg (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| RqstID (XML)<br><br>rqstID (JSON) | CHAR 10 | Y | | | Y | Group433 | Unique Reference Number used for identifying a Group Request application. |
| OutstationID (XML)<br><br>outstationID (JSON) | CHAR 09 | Y | | | Y | OUT11 | Airlines Outstation/Sales office where request originates from. |
| RequestDate (XML)<br><br>requestDate (JSON) | DATE | Y | | | Y | 20050930 | Date group booking request submitted to G6.<br>Format: YYYYMMDD |
| RequestTime (XML)<br><br>requestTime (JSON) | TIME 04 | Y | | | Y | 1130 | Time group booking request submitted to G6.<br>Format: 0000 – 2359 |
| <Note> (XML)<br><br>"note" (JSON) | | Y | | | | | Reject Note information |

##### *3.1.2.5.2.2 Note Information*

There can be only one Note record for each request. The Note record can be used to pass information on the reason for rejection.

RecordSet: Note (for XML) and note (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Type (XML)<br><br>type (JSON) | CHAR 04 | Y | | | Y | REJ | The reject note identifier. Use REJ for the rejection. |
| Value (XML)<br><br>value (JSON) | CHAR 256 | Y | | | | | The text content for rejection |

#### 3.1.2.5.1 Process Rules

None

#### 3.1.2.5.2 Loader Behavior

Insert/Update

#### 3.1.2.5.3 Schema

The content of schema can be found at /config/interface/xsd/reject.xsd file.

#### 3.1.2.5.4 Examples

```
<?xml version="1.0" ?>
<RejectLoad>
  <RJG>
    <RqstID>Group433</RqstID>
```

```
    <OutstationID>OUT11</OutstationID>
    <RequestDate>20050930</RequestDate>
    <RequestTime>1130</RequestTime>
     <Note>
           <Type>REJ</Type>
           <Value>This request is rejected by travel agent John due to the low offer
              fare</Value>
        </Note>
  </RJG>
</RejectLoad>
```

#### 3.1.2.5.5 Reject Json load via Reject rest loader

Endpoint: **/group/rest/submitRejectRequest**

Example: http://localhost:8060/group/rest/submitRejectRequest

#### 3.1.2.5.6 Example reject JSON content for Reject rest loader

```
{
   "rjg":{
      "rqstID":"A1094",
      "outstationID":"homeoffic",
      "requestDate":"20180618",
      "requestTime":"1508",
      "note":{
         "type":"REJ",
         "value":"rejected with some reason"
      }
   }
}
```

### *System Outputs*

### 3.1.3 Output Dataset Record Definitions

The system will produce the following datasets. More details can be found in the section below for each of the data outputs.

#### *3.1.3.1 Negotiation*

##### 3.1.3.1.1 Description

The negotiation module has a temporary holding file where the alternative options are stored by the PROS Group System until they are sent back to the Outstation for review. Unlike other queues in the Group System, the negotiation module generates an external file, which is picked up by Airlines' interface so that the contents can be sent to the Mask. The name of the file generated is "RequestID"_HLD.xml, and it contains all of the alternative options for one group request. The file is generated in the directory from which the Group System was started. Please note that as the file is processed by Airline's interface, it is not removed from the directory where was generated

##### 3.1.3.1.2 Data Fields

This record contains general request level information. There will be one HHG record for each request.

DatasetName: HoldingOption
RecordSet: HHG (for XML) and hhg (for JSON response)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| RqstID (XML)<br><br>rqstID (JSON) | CHAR 10 | Y | | | | Group433 | Unique Reference Number used for identifying a Group Request application. |
| OutstationID (XML)<br><br>outstationID (JSON) | CHAR 09 | Y | | | | OUT11 | Airlines Outstation/Sales office where request originates from. |
| RequestDate (XML)<br><br>requestDate (JSON) | DATE | Y | | | | 20050930 | Date group booking request submitted to G6.<br>Format: YYYYMMDD |
| RequestTime (XML)<br><br>RequestTime (JSON) | TIME 04 | Y | | | | 1130 | Time group booking request submitted to G6.<br>Format: 0000 – 2359 |
| GroupName (XML)<br><br>groupName (JSON) | CHAR 50 | Y | | | | U OF H GROUP | Name of the group for use in PNR Creation. |
| CustomerType (XML)<br><br>CustomerType (JSON) | NUM 10 | Y | Customer type | | | 3 | Number used to identify category type (e.g.: Leisure, Cruise, Sports, etc.) |
| AgencyIata (XML)<br><br>agencyIata (JSON) | CHAR 10 | N | Agency | | | A987654321 | GDS or other Airlines IATA reference. Don't pass this field if there is no value. |
| OfficeCode (XML)<br><br>officeCode (JSON) | CHAR 10 | N | Agency | | | FRALH0860 | Office Code.<br>Either IATA or Office Code must exist. Don't pass this field if there is no value. |
| AgencyName (XML)<br><br>agencyName (JSON) | CHAR 50 | N | Agency | | | Discount Travelers | GDS or other Airlines Name. Don't pass this field if there is no value. |
| GroupType (XML)<br><br>groupType (JSON) | CHAR 02 | Y | | | | A | A = Ad hoc<br>S = Series |
| CityPOS (XML)<br><br>cityPOS (JSON) | CITY | N | | | | IAH | Departure Airport for the City where the request originated.<br>If blank, will use info from Agency table.<br>In case of new Agency IATA, this field will populate the City POS field in the Agency table in G6. Don't pass this field if there is no value. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| CommPnct (XML)<br><br>commPnct (JSON) | NUM 02 | Y | | | | 0 | Agent commission percentage.<br>Must be a whole number. |
| SeatsRqstd (XML)<br><br>seatsRqstd (JSON) | NUM 03 | Y | | | | 20 | |
| SeatsPasses (XML)<br><br>seatsPasses (JSON) | NUM 03 | Y | | | | 0 | Non-revenue tickets. |
| SeatsExp (XML)<br><br>seatsExp (JSON) | NUM 03 | Y | | | | 16 | Seats expected. |
| MinSeatRqstd (XML)<br><br>minSeatRqstd (JSON) | NUM 03 | N | | | | 20 | This value will be used only if<br>Passenger negotiation is set (NegoFlag )<br>Its value can only be smaller than the seats requested.<br>If this value is not passed and the NegoFlag is set to Passenger (4), the request will be rejected. |
| QuotedFare (XML)<br><br>quotedFare (JSON) | NUM 10 | Y | | | | 500 | Fare originally quoted for the request |
| RealFare (XML)<br><br>realFare (JSON) | NUM 10 | N | | | | 500 | Fare accepted in the G6 for the request. By default, it has the same value as quoted fare. Don't pass this field if there is no value. |
| NegoFlag (XML)<br><br>negoFlag (JSON) | NUM 01 | Y | | | | 1 | 1: Flight negotiation<br>2: Fare negotiation (mainly for the preferred flight)<br>4: Passenger negotiation (mainly for the preferred flight)<br>3: Fare and Flight negotiation (mainly for the preferred flight) |
| UtlRate (XML)<br><br>utlRate (JSON) | NUM 03 | Y | | | | 80 | Utilization rate |
| ProgramName (XML)<br><br>programName (JSON) | CHAR 50 | N | | | | | Program name |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| LocalFare (XML)<br><br>localFare (JSON) | NUM 10 | N | | | | 500 | Fare being charged per person in Local Currency |
| <HFG> (XML)<br><br>"hfg" (JSON) | | Y | | | | | Direction information |
| <Note> (XML)<br><br>"note" (JSON) | | Y | | | | | Note information |

#### *3.1.3.1.2.1 Direction Information*

The HFG record identifies the direction that is being held; therefore there can be only one record for each requested direction.

RecordSet: HFG (for XML ) and hfg (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DirNum (XML)<br><br>dirNum (JSON) | NUM 02 | Y | | | | 1 | Direction number starting from 1. The outbound direction must be direction 1. |
| Orgn (XML)<br><br>orgn (JSON) | CITY | Y | | | | IAH | Direction origin |
| Dstn (XML)<br><br>dstn (JSON) | CITY | Y | | | | FRA | Direction destination |
| SumFareCase (XML)<br><br>sumFareCase (JSON) | CHAR 02 | Y | | | | CW | Sum of marginal fares for each segment.<br>The Sum Fare information will be populated either from the request information or the system default. |
| DenyBoardingFactor (XML)<br><br>denyBoardingFactor (JSON) | CHAR 02 | Y | | | | | Factor used to load marginal fare according to Fight/date demand levels.<br>The Denied Boarding information will be populated either from the request information or the system default. |
| RqstdCmp (XML)<br><br>rqstdCmp (JSON) | SYMBOL 02 | Y | | | | M | Compartment code identifying requested cabin. |
| TripDuration (XML)<br><br>tripDuration (JSON) | NUM 03 | N | | | | 5 | Trip duration number. The number not applies to the first direction. If there is not value, do not send this attribute out. |
| LocalFare (XML)<br><br>localFare (JSON) | NUM 10 | N | | | | 500 | Fare being charged per person in Local Currency. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| <HFO> (XML)<br><br>"hfo" (JSON) | | Y | | | | | Option information |

#### *3.1.3.1.2.2 Note Information*

RecordSet: Note (for XML) and note (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Type (XML)<br><br>type (JSON) | CHAR 04 | Y | | | Y | CTC | The identifier for note text. There is no restriction of unique note type. |
| Value (XML)<br><br>value (JSON) | CHAR 256 | Y | | | | | The text content |

#### *3.1.3.1.2.3 Option Information*

The HFO record contains information on the option(s) offered for a direction. Since there can be one HFO record for each option offered and it is possible to offer multiple options, the file can contain multiple HFO records.

RecordSet:HFO (for XML) and hfo (for JSON)

| Data Field | Format | Rq'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| OptID (XML)<br><br>optID (JSON) | NUM 04 | Y | | | | 1 | Option ID. |
| SubOptID (XML)<br><br>subOptID (JSON) | NUM 04 | Y | | | | 1 | Sub Option ID. When Sub Option is not the same with Option ID, it indicates option break up. |
| DptDate (XML)<br><br>dptDate (JSON) | DATE | Y | | | | 20051031 | Departure Date |
| DptTime (XML)<br><br>dptTime (JSON) | TIME 04 | Y | | | | 1030 | Departure Time |
| ArrTime (XML)<br><br>arrTime (JSON) | TIME 04 | Y | | | | 1930 | Arrival Time |
| ArrDayOffset (XML)<br><br>arrDayOffset (JSON) | NUM 02 | N | | | | | 0: If the arrival date and departure date are the same.<br>1: If the arrival date is one day after the departure date.<br>2: If the arrival date is two days after the departure date. |
| SegOrgn (XML)<br><br>segOrgn (JSON) | CITY | Y | | | | FRA | Segment Origin |
| SegDstn (XML)<br><br>segDstn (JSON) | CITY | Y | | | | JFK | Segment Destination |
| NormAvail (XML)<br><br>normAvail (JSON) | NUM 03 | Y | | | | 125 | Normalized availability |

| Data Field | Format | Rq'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| SeatsRqstd (XML)<br><br>seatsRqstd (JSON) | NUM 03 | Y | | | | 100 | Seat requested for this option. |
| MargFare (XML)<br><br>margFare (JSON) | NUM 10 | Y | | | | 500 | Marginal fare |
| CmpCode (XML)<br><br>cmpCode (JSON) | SYMBOL 02 | N | | | | M | Highest Compartment code from flights. |
| ClassCode (XML)<br><br>classCode (JSON) | SYMBOL 02 | Y | | | | M | Class symbol for this option associated with Highest Compartment code |
| OptSelected (XML)<br><br>optSelected (JSON) | NUM 01 | Y | | | | 1 | 1 = Selected |
| ClassPOS (XML)<br><br>classPOS (JSON) | CHAR 10 | N | | | | M | |
| RealFare (XML)<br><br>realFare (JSON) | NUM 10 | Y | | | | | Real fare |
| FlightInfo (XML)<br><br>flightInfo (JSON) | CHAR 128 | Y | | | | | Carrier code and flight number. If option has more than one flight then each flight info will be separated by '/' symbol. |
| ConnectionCities (XML)<br><br>connectionCities (JSON) | CHAR 11 | N | | | | | If option has more than one flights then the connection city(s) are listed here separated by '/' symbol |
| PreferredOption (XML)<br><br>preferredOption (JSON) | Flag 01 | N | | | | 1 | 1: This option contains preferred flight.<br>0: no preferred flight in this option. |
| LocalFare (XML)<br><br>localFare (JSON) | NUM 10 | N | | | | 500 | Fare in Local Currency. |
| <HFF> (XML)<br><br>"hff" (JSON) | | Y | | | | | Segment information |

#### *3.1.3.1.2.4 Option Segment/Leg Information*

There can be multiple HFF records. Each record contains the information for each traveled flight segment (segment control carriers) constituting each itinerary option. Most of the fields are provided except the ExpEmpty (not available and thus 0 for the leg control carriers). For the leg inventory control carriers, the leg will replace segment information. Each flight/leg will have one HFF record.

RecordSet: HFF (for XML) and hff (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| CrrCode (XML)<br><br>crrCode (JSON) | CARRIER | Y | | | | ZZ | Carrier Code |
| FlightNum (XML)<br><br>flightNum (JSON) | FLIGHT 05 | Y | | | | 441 | Flight number |
| DptDate (XML)<br><br>dptDate (JSON) | DATE | Y | | | | 20051031 | Format:<br>YYYYMMDD. |
| SegOrgn (XML)<br><br>segOrgn (JSON) | CITY | Y | | | | IAH | Travel Segment Origin for segment control carriers.<br><br>Travel Leg Origin for the leg inventory control carriers. |
| SegDstn (XML)<br><br>segDstn (JSON) | CITY | Y | | | | FRA | Travel Segment Destination for segment control carriers.<br><br>Travel Leg Destination for the leg inventory control carriers. |
| DptTime (XML)<br><br>dptTime (JSON) | TIME 04 | Y | | | | 1030 | Departure Time |
| ArrTime (XML)<br><br>arrTime (JSON) | TIME 04 | Y | | | | 1930 | Arrival Time |
| ActAvail (XML)<br><br>actAvail (JSON) | NUM 03 | Y | | | | 125 | Actual availability |
| NormAvail (XML)<br><br>normAvail (JSON) | NUM 03 | Y | | | | 125 | Normalized availability |
| ExpEmpty (XML)<br><br>expEmpty (JSON) | NUM 03 | Y | | | | 10 | Segment expected empty. |
| TotalBook (XML)<br><br>totalBook (JSON) | NUM 03 | Y | | | | 200 | Segment total book |
| GrpBook (XML)<br><br>grpBook (JSON) | NUM 03 | Y | | | | 100 | Segment group book |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| MargFare (XML)<br><br>margFare (JSON) | NUM 10 | Y | | | | 1 | Marginal fare |
| InProsFlag (XML)<br><br>inProsFlag (JSON) | FLAG 01 | Y | | | | | 1= Flight is contained in the PROS database (default)<br>0 = Flight is not contained in the PROS database |
| ClassCode (XML)<br><br>classCode (JSON) | SYMBOL 02 | N | | | | M | Class symbol for this flight |
| ClassPOS (XML)<br><br>classPOS (JSON) | CHAR 10 | N | | | | ALL | Class Point of Sale |

#### 3.1.3.1.3 Process Rules

None

#### 3.1.3.1.4 Loader Behavior

None

#### 3.1.3.1.5 Schema

The content of schema can be found at /config/interface/xsd/nego.xsd file.

#### 3.1.3.1.6 Examples

```
<?xml version="1.0" ?>
<HoldingOption>
  <HHG>
   <RqstID>Group433</RqstID>
   <OutstationID>OUT11</OutstationID>
   <RequestDate>20050930</RequestDate>
   <RequestTime>1130</RequestTime>
   <GroupName>U OF H GROUP</GroupName>
   <CustomerType>3</CustomerType>
   <AgencyIata>A987654321</AgencyIata>
   <OfficeCode>FRALH0860</OfficeCode>
   <AgencyName>Discount Travellers</AgencyName>
   <GroupType>A</GroupType>
   <CityPOS>IAH</CityPOS>
   <CommPnct>0</CommPnct>
   <SeatsRqstd>20</SeatsRqstd>
   <SeatsPasses>0</SeatsPasses>
   <SeatsExp>0</SeatsExp>
   <MinSeatRqstd> 20</MinSeatRqstd>
   <QuotedFare>500</QuotedFare>
   <RealFare>500</RealFare>
   <NegoFlag>1</NegoFlag>
   <UtlRate>70</UtlRate>
   <Note>
      <Type>CTC</Type>
```

```
      <Value> Boris Walden</Value>
   </Note>
   <Note>
      <Type>RCV</Type>
      <Value> Jurgen Singer </Value>
     </Note>
 <Note>
   <Type>SSR</Type>
   <Value> SSR value </Value>
   </Note>
   <Note>
      <Type>OSI</Type>
      <Value> OSI value</Value>
   </Note>
   <Note>
      <Type>IMP</Type>
      <Value>IMP value</Value>
   </Note>
   <Note>
      <Type>HLD</Type>
      <Value>HLD value</Value>
   </Note>
   <Note>
      <Type>STO</Type>
      <Value>This is a value to be stored</Value>
   </Note>
   <Note>
      <Type>TXT</Type>
      <Value>This is a PNR Remark</Value>
   </Note>
   <HFG>
      <DirNum>1</DirNum>
      <Orgn>JFK</Orgn>
      <Dstn>DEL</Dstn>
      <SumFareCase>SF</SumFareCase>
      <DenyBoardingFactor>DB</DenyBoardingFactor>
      <RqstdCmp>M</RqstdCmp>
     <HFO>
      <OptID>1</OptID>
      <SubOptID>1</SubOptID>
      <DptDate>20050913</DptDate>
      <DptTime>1030</DptTime>
      <ArrTime>0400</ArrTime>
      <SegOrgn>LHR</SegOrgn>
      <SegDstn>JFK</SegDstn>
      <NormAvail>50</NormAvail>
      <SeatsRqstd>25</SeatsRqstd>
      <MargFare>550</MargFare>
      <ClassCode>Y</ClassCode>
      <OptSelected>1</OptSelected>
      <ClassPOS>USA</ClassPOS>
      <RealFare>600</RealFare>
      <FlightInfo>ZZ132/ZZ133</FlightInfo>
      <ConnectionCities>LHR</ConnectionCities>
      <HFF>
         <CrrCode>ZZ</CrrCode>
         <FlightNum>132</FlightNum>
```

```
            <DptDate>20050913</DptDate>
            <SegOrgn>JFK</SegOrgn>
            <SegDstn>LHR</SegDstn>
            <DptTime>1030</DptTime>
            <ArrTime>0600</ArrTime>
            <ActAvail>50</ActAvail>
            <NormAvail>50</NormAvail>
            <ExpEmpty>35</ExpEmpty>
            <TotalBook>180</TotalBook>
            <GrpBook>30</GrpBook>
            <MargFare>550</MargFare>
            <InProsFlag>1</InProsFlag>
            <ClassCode>G</ClassCode>
         </HFF>
         <HFF>
            <CrrCode>ZZ</CrrCode>
            <FlightNum>133</FlightNum>
            <DptDate>20050914</DptDate>
            <SegOrgn>LHR</SegOrgn>
            <SegDstn>DEL</SegDstn>
            <DptTime>1200</DptTime>
            <ArrTime>0400</ArrTime>
            <ActAvail>72</ActAvail>
            <NormAvail>72</NormAvail>
            <ExpEmpty>48</ExpEmpty>
            <TotalBook>160</TotalBook>
            <GrpBook>43</GrpBook>
            <MargFare>480</MargFare>
            <InProsFlag>1</InProsFlag>
            <ClassCode>G</ClassCode>
         </HFF>
      </HFO>
   </HFG>
   </HHG>
</HoldingOption>
```

#### 3.1.3.1.7 Json output for Offer/Negotiation loader in GroupAgent(MASK)

Mask Endpoint to load the offer: **/groupagent/rest/offerRequest**
Exmple Mask endpoint: http://localhost:8090/groupagent/rest/offerRequest

#### 3.1.3.1.8 Offer JSON output example

```
{
    "hhg": {
        "rqstID": "A756295",
        "outstationID": "homeoffic",
        "requestDate": "20180628",
        "requestTime": "1536",
        "groupName": "TEST",
        "customerType": "2",
        "agencyIata": "38519644",
        "agencyName": "AAA",
        "groupType": "A",
        "cityPOS": "CDG",
        "commPnct": "0",
        "seatsRqstd": "10",
        "seatsPasses": "0",
```

```
        "seatsExp": "10",
        "minSeatRqstd": "0",
        "quotedFare": "0",
        "realFare": "1000",
        "negoFlag": "3",
        "utlRate": "100",
        "localFare": "7",
        "note": [{
            "type": "RCV",
            "value": "rejuu"
        },
        {
            "type": "NEG",
            "value": "Default negotiation message from UI"
        }],
        "hfg": [{
            "dirNum": "1",
            "orgn": "CDG",
            "dstn": "DOH",
            "sumFareCase": "CW",
            "denyBoardingFactor": "Y",
            "rqstdCmp": "Y",
            "localFare": "7",
            "hfo": [{
                "optID": "1",
                "subOptID": "1",
                "dptDate": "20181212",
                "dptTime": "2120",
                "arrTime": "0540",
                "arrDayOffset": "1",
                "segOrgn": "CDG",
                "segDstn": "DOH",
                "normAvail": "263",
                "seatsRqstd": "10",
                "margFare": "44",
                "cmpCode": "Y",
                "classCode": "G",
                "optSelected": "1",
                "realFare": "1000",
                "flightInfo": "QR00038",
                "localFare": "7",
                "hff": [{
                    "crrCode": "QR",
                    "flightNum": "00038",
                    "dptDate": "20181212",
                    "segOrgn": "CDG",
                    "segDstn": "DOH",
                    "dptTime": "2120",
                    "arrTime": "0540",
                    "actAvail": "209",
                    "normAvail": "209",
                    "expEmpty": "197",
                    "totalBook": "0",
                    "grpBook": "26",
                    "margFare": "44",
                    "inProsFlag": "1",
                    "classCode": "G"
                }]
            }]
        }]
    }
}
```

### *3.1.3.2 Confirmation/Rejection (Deprecated)*

This record structure is deprecated and will be removed in a future release. It will be replaced by the new Confirmation/Rejection record structure defined in 3.1.3.3.

#### 3.1.3.2.1 Description

PROS Group System provides the ability to send the confirmation information to the original requester when the group request has been accepted. In similar manner, the PROS Group System will allow the user to send a rejection remark to the requested party. Only one confirmation or rejection message is allowed per request.

#### 3.1.3.2.2 Data Fields

This REQ record contains the Request ID to be able to reference the message to the appropriate request.

RecordSet: REQ

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| RqstID | CHAR 10 | Y | | | | 1011000530 | Request ID |
| OutstationID | CHAR 09 | N | | | | OUT11 | Outstation ID. Don't pass this field if there is no value. |
| RequestDate | DATE | N | | | | 20050930 | Request Date. Don't pass this field if there is no value. |
| RequestTime | TIME 04 | N | | | | 1130 | Request Time. Don't pass this field if there is no value. |
| <Note> | | N | | | | | Note information. Don't pass this field if there is no value. |
| <Loc> | | N | | | | | Locator information.<br>This information is provided only when the request is confirmed. Don't pass this field if there is no value. |

##### *3.1.3.2.2.1 Note Data Fields*

This record is the message sending back to the mask system. It is optional.

RecordSet: Note

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Type | CHAR 04 | Y | | | | RJT CFM, …TXT | RJT : the rejection (set as the default and can be modified)<br>CFM: confirmation (Set as default and can be modified)<br>TXT and other possible note message can be passed back to the outstations.<br>A reserved ERR is to indicate any error message provided by G6. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Value | CHAR 255 | N | | | | Example1. Thank you for your business.<br><br>Example2. Not enough Quoted Fare. | The message going out to the outstations. Don't pass this field if there is no value. |

#### *3.1.3.2.2.2 Locator Data Fields*

If this is a confirmation message, this Loc record will contain PNR Locator information. In case of rejection this record may be omitted.

RecordSet: Loc

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| PNRLocator | LOCATOR | Y | | | | AAAAAA20050930 | PNR Locator |
| PttnSequence | NUM 02 | N | | | | 1 | Series only |
| DptSequence | NUM 03 | N | | | | | Series Only |
| BkgStatus | SYMBOL 02 | N | | | | | Series Only |
| ProcessId | NUM 10 | N | | | | | Process ID sent back to the client. |
| GroupShellID | NUM 38 | N | | | | | This field is designed specifically for the Kebut interface. |

#### *3.1.3.2.2.3 Term Data Fields*

The data fields contain necessary to support terms and conditions.

RecordSet: TERMS

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Id | NUM 38 | N | | | | | Either Group shell ID or Series ID |
| MaxGrpSizeTot | NUM 04 | Y | | | | | Allocations Total Group Size/Max as per Deal Note |
| ClassCode | SYMBOL 02 | Y | | | | | Booking Class AS per request |
| LocalFare | NUM 10 | Y | | | | | Net Fare As per Deal Code Message Net Fare |
| FareBasis | CHAR 8 | Y | | | | | FARE BASIS As per Deal Note |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| DealDaysPrior | Num 04 | Y | | | | | From Date of Deal to TTL. Here the TTL is assumed to be the first valid date in TTL. If Deal is done of Jan 1, 2007 and Ticketing Time Limit for deposit is 30 days prior, this value will be 30. |
| TravelDaysPrior | NUM 04 | Y | | | | | TRAVEL VALIDITY From 1st travel date to last travel date. Difference between the departure date of first direction and the departure date of last direction. So for 1 direction Group Request, this can be 0. |
| MinGrpSizeTot | NUM 04 | Y | | | | | Minimum 104equested104n (Allocations) Total Group Size/Min as per Deal Note |
| Deposit | NUM 10 | Y | | | | | Deposit 25% of total group fare |
| MinGrpSizeFlt | NUM 04 | Y | | | | | Minimum 104tilization per flight As per Deal Code Message Min Group Size per flight |
| DepositDeadlineDate | DATE | Y | | | | | Release Period if deposit not paid. 8 days after PNR confirmation |
| NamesDeadlineDate | DATE | Y | | | | | Deadline for names. AS per G6 Name Time Line |
| TtlDate | DATE | Y | | | | | Ticketing time limit. As per G6 TTL |
| StopOvrCost | NUM 10 | Y | | | | | STPC |
| StopOver | CHAR 11 | Y | | | | | TRAVEL VALIDITY |
| Endorsement | CHAR 11 | Y | | | | | Endorsement |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| FareBox | CHAR 11 | Y | | | | | Fare box |
| Commission | CHAR 11 | Y | | | | | Commission |
| Application | CHAR 11 | Y | | | | | Application |
| SalesRestrictions | CHAR 11 | Y | | | | | Sales restrictions |
| Reservations | CHAR 11 | Y | | | | | Reservations |
| Combinations | CHAR 11 | Y | | | | | Combinations |
| SideTrips | CHAR 11 | Y | | | | | Sidetrips |
| Transfers | CHAR 11 | Y | | | | | Transfers |
| ReservationChange | CHAR 11 | Y | | | | | Change of reservation |
| NameChangePre | CHAR 11 | Y | | | | | Name change before ticketing |
| NameChangePost | CHAR 11 | Y | | | | | Name change after ticketing |
| LostTicketFee | CHAR 11 | Y | | | | | Lost ticket fee |
| CancellationFee | CHAR 11 | Y | | | | | Cancellation/Refund |
| NoShowFee | CHAR 11 | Y | | | | | No show fee |
| ChildFare | CHAR 11 | Y | | | | | Child fare |
| InfantFare | CHAR 11 | Y | | | | | Infant fare |

#### 3.1.3.2.3 Processing Rules

None

#### 3.1.3.2.4 Output Behavior

None.

#### 3.1.3.2.5 Schema

The content of schema can be found at /config/interface/xsd/confrej.xsd file.

#### 3.1.3.2.6 Example (Confirmation)

```
<?xml version="1.0" ?>
<Reply>
  <REQ>
   <RqstID>1011000530</RqstID>
   <OutstationID>OUT11</OutstationID>
   <RequestDate>20050930</RequestDate>
   <RequestTime>1130</RequestTime>
   <Note>
      <Type>CTC</Type>
      <Value> Boris Walden</Value>
   </Note>
   <Note>
```

```
      <Type>RCV</Type>
      <Value> Jurgen Singer </Value>
   </Note>
   <Note>
      <Type>TXT</Type>
      <Value>We appreciate your business.</Value>
   </Note>
   <Loc>
     <PNRLocator>AAAAAA20050930</PNRLocator>
   </Loc>
   <Loc>
     <PNRLocator>BBBBBB20050930</PNRLocator>
     </Loc>
  </REQ>
</Reply>
```

#### 3.1.3.2.7 Example (Rejection)

```
<?xml version="1.0" ?>
<Reply>
  <REQ>
   <RqstID>1011000530</RqstID>
   <OutstationID>OUT11</OutstationID>
   <RequestDate>20050930</RequestDate>
   <RequestTime>1130</RequestTime>
   <Note>
      <Type>CTC</Type>
      <Value> Boris Walden</Value>
   </Note>
   <Note>
      <Type>RCV</Type>
      <Value> Jurgen Singer </Value>
   </Note>
   <Note>
      <Type>TXT</Type>
      <Value>Not enough Quoted Fare.</Value>
   </Note>
  </REQ>
</Reply>
```

### *3.1.3.3 Confirmation/Rejection*

#### 3.1.3.3.1 Description

PROS Group System provides the ability to send the confirmation information to the original requester when the group request has been accepted. In similar manner, the PROS Group System will allow the user to send a rejection remark to the requested party. Only one confirmation or rejection message is allowed per request.

#### 3.1.3.3.2 Data Fields

This REQ record contains the Request ID to be able to reference the message to the appropriate request.

RecordSet: REQ (for XML) and req (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| RqstID (XML)<br>rqstID (JSON) | CHAR 10 | Y | | | | 1011000530 | Request ID |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| OutstationID (XML)<br><br>outstationID (JSON) | CHAR 09 | N | | | | OUT11 | Outstation ID. Don't pass this field if there is no value. |
| RequestDate (XML)<br><br>requestDate (JSON) | DATE | N | | | | 20050930 | Request Date. Don't pass this field if there is no value. |
| RequestTime (XML)<br><br>requestTime (JSON) | TIME 04 | N | | | | 1130 | Request Time. Don't pass this field if there is no value. |
| <Note> (XML)<br><br>"note" (JSON) | | N | | | | | Note information. Don't pass this field if there is no value. |
| <AdHoc> (XML)<br><br>"adhoc" (JSON) | | N | | | | | AdHoc information. There can be only one of these. Only AdHoc or Pattern can be defined, not both. |
| <Pattern> (XML)<br><br>"pattern" (JSON) | | N | | | | | Pattern information for Series Requests. Can have more than one pattern per Series. Only AdHoc or Pattern can be defined, not both. |

#### *3.1.3.3.2.1 Note Data Fields*

This record is the message sending back to the mask system. It is optional.

RecordSet: Note (for XML) and note (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Type (XML)<br><br>type (JSON) | CHAR 04 | Y | | | | RJT CFM, …TXT | RJT : the rejection (set as the default and can be modified)<br>CFM: confirmation (Set as default and can be modified)<br>TXT and other possible note message can be passed back to the outstations.<br>A reserved ERR is to indicate any error message provided by G6. |
| Value (XML)<br><br>value (JSON) | CHAR 255 | N | | | | Example1. Thank you for your business.<br><br>Example2. Not enough Quoted Fare. | The message going out to the outstations. Don't pass this field if there is no value. |

#### *3.1.3.3.2.2 Pattern Data Fields*

This will contain the data for the AdHoc Request that is being confirmed. This data will include the Locators and Terms if any.

RecordSet: Pattern (for XML) and pattern (for JSON)

**Attributes (only for XML)**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| pttnSequence (only for XML) | NUM 02 | Y | | | | 1 | The pattern number in the series.<br>For XML response, this field comes as attribute for Patten. |

**Elements**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| pttnSequence (only for JSON) | NUM 02 | Y | | | | 1 | The pattern number in the series.<br>For JSON response, this field comes under pattern, but not as attribute |
| <Loc> (XML)<br>"loc" (JSON) | | Y | | | | | Locator information.<br>This information is provided only when the request is confirmed. Don't pass this field if there is no value. |

#### *3.1.3.3.2.3 AdHoc Data Fields*

This will contain the data for the AdHoc Request that is being confirmed. This data will include the Locators and Terms if any.

RecordSet: AdHoc (for XML) and adHoc (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| <Loc> (XML)<br>"loc" (JSON) | | Y | | | | | Locator information.<br>This information is provided only when the request is confirmed. Don't pass this field if there is no value. |

#### *3.1.3.3.2.4 Locator Data Fields*

If this is a confirmation message, this Loc record will contain PNR Locator information. In case of rejection this record may be omitted.

RecordSet: Loc (for XML) and loc (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| PNRLocator (XML)<br><br>pnrLocator (JSON) | LOCATOR | Y | | | | AAAAAA20050930 | PNR Locator |
| DptSequence (XML)<br><br>dptSequence (JSON) | NUM 03 | Y/N | | | | | Required for Series, leave blank for AdHoc |
| BkgStatus (XML)<br><br>bkgStatus (JSON) | SYMBOL 02 | Y/N | | | | | Required for Series, leave blank for AdHoc |
| ProcessId (XML)<br><br>processId (JSON) | NUM 10 | N | | | | | Process ID sent back to the client. |
| GroupShellID (XML)<br><br>groupShellID (JSON) | NUM 38 | N | | | | | This field is designed specifically for the Kebut interface. |
| <Terms> (XML)<br><br>"terms" (JSON) | | N | | | | | Optional list of Terms and Conditions |

### *3.1.3.3.2.5 Terms and Conditions field*

This is the list of terms and conditions.

RecordSet: Terms (for XML) and terms (for JSON)

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| <Term> (XML)<br><br>"term" (JSON) | CHAR 255 | Y | | | | | Each Term or Condition will have one <Term> entry. |

**Term attributes (only for XML)**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| key | CHAR 32 | Y | | | | | Key for a Term or Condition |

**RecordSet: term (only for JSON)**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| key | CHAR 32 | Y | | | | Fare Basis | Key for a Term or Condition |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| value | CHAR 255 | Y | | | | JFRG00F | Value for a term or condition |

#### 3.1.3.3.3 Processing Rules

None

#### 3.1.3.3.4 Output Behavior

None.

#### 3.1.3.3.5 Schema

The content of schema can be found at /config/interface/xsd/confirmreject.xsd file.

#### 3.1.3.3.6 Example (AdHoc Confirmation) (XML based confirmation)

```
<?xml version="1.0" ?>
<Reply>
  <REQ>
   <RqstID>1011000530</RqstID>
   <OutstationID>OUT11</OutstationID>
   <RequestDate>20050930</RequestDate>
   <RequestTime>1130</RequestTime>
   <Note>
      <Type>CTC</Type>
      <Value> Boris Walden</Value>
   </Note>
   <Note>
      <Type>RCV</Type>
      <Value> Jurgen Singer </Value>
   </Note>
   <Note>
      <Type>TXT</Type>
      <Value>We appreciate your business.</Value>
   </Note>
   <AdHoc>
      <Loc>
        <PNRLocator>AAAAAA20050930</PNRLocator>
        <Terms>
          <Term key="Term1">Value1</Term>
          <Term key="Term2">Value2</Term>
        </Terms>
      </Loc>
      <Loc>
        <PNRLocator>BBBBBB20050930</PNRLocator>
        <Terms>
          <Term key="Term1">Value1</Term>
          <Term key="Term2">Value2</Term>
        </Terms>
      </Loc>
   </AdHoc>
  </REQ>
</Reply>
```

### 3.1.3.3.7 Example (AdHoc Confirmation) (JSON based confirmation)

Mask endpoint**: /groupagent/rest/acceptRequest**

Exmple: http://localhost:8090/groupagent/rest/acceptRequest

```
{
    "req": {
        "rqstID": "A756296",
        "outstationID": "homeoffic",
        "requestDate": "20180628",
        "requestTime": "1536",
        "note": [{
            "type": "RCV",
            "value": "rejuu"
        },
        {
            "type": "CFM",
            "value": "Confirmed"
        }],
        "adHoc": {
            "loc": [{
                "pnrLocator": "QKRRSK/OKCASJ20170101",
                "terms": [{
                    "key": "Group Shell",
                    "value": "66801"
                },
                {
                    "key": "Deal Date",
                    "value": "20170101"
                },
                {
                    "key": "Fare Basis",
                    "value": "JFRG00F"
                }]
            }]
        }
    }
}
```

### 3.1.3.3.8 Example (Series Confirmation) (XML based confirmation)

```
<?xml version=”1.0” ?>
<Reply>
  <REQ>
   <RqstID>1011000530</RqstID>
   <OutstationID>OUT11</OutstationID>
   <RequestDate>20050930</RequestDate>
   <RequestTime>1130</RequestTime>
   <Note>
      <Type>CTC</Type>
      <Value> Boris Walden</Value>
   </Note>
   <Note>
      <Type>RCV</Type>
      <Value> Jurgen Singer </Value>
   </Note>
   <Note>
      <Type>TXT</Type>
      <Value>We appreciate your business.</Value>
   </Note>
   <Pattern pttnSequence=”1”>
```

```
      <Loc>
        <PNRLocator>AAAAAA20050930</PNRLocator>
        <Terms>
           <Term key=”Term1”>Value1</Term>
           <Term key=”Term2”>Value2</Term>
        </Terms>
      </Loc>
      <Loc>
        <PNRLocator>BBBBBB20050930</PNRLocator>
        <Terms>
           <Term key=”Term1”>Value1</Term>
           <Term key=”Term2”>Value2</Term>
        </Terms>
      </Loc>
   </Pattern>
   <Pattern pttnSequence=”2”>
      <Loc>
        <PNRLocator>CCCCCC20050930</PNRLocator>
        <Terms>
           <Term key=”Term1”>Value1</Term>
           <Term key=”Term2”>Value2</Term>
        </Terms>
      </Loc>
      <Loc>
        <PNRLocator>DDDDDD20050930</PNRLocator>
        <Terms>
           <Term key=”Term1”>Value1</Term>
           <Term key=”Term2”>Value2</Term>
        </Terms>
      </Loc>
   </Pattern>
  </REQ>
</Reply>
```

#### 3.1.3.3.9 Example (Series Confirmation) (JSON based confirmation)

Mask endpoint**: /groupagent/rest/acceptRequest**

Exmple: http://localhost:8090/groupagent/rest/acceptRequest

```
{
   "req": {
      "rqstID": "S918",
      "outstationID": "LISTP0550",
      "requestDate": "20170724",
      "requestTime": "1149",
      "note": [{
         "Type": "CFM",
         "Value": "Confirmed"
      }],
      "pattern": [{
         "pttnSequence": "1",
         "loc": [
            {
               "pnrLocator": "YJNHBQ/RESJQL20170101",
               "dptSequence": "1",
               "bkgStatus": "HK",
               "processId": "0",
               "terms": [
                  {
                     "key": "Additional Conditions",
                     "value": "ATTACH SUPPORTING DOCUMENTS TO THE AUDIT COUPON"
```

```
      },
      {
         "key": "Baggage",
         "value": "APPLY NORMAL BAGGAGE ALLOWANCE"
      }
   ]
},
{
   "pnrLocator": "KORPBK/HFDGLO20170101",
   "dptSequence": "2",
   "bkgStatus": "HK",
   "processId": "0",
   "terms": [
      {
         "key": "Additional Conditions",
         "value": "ATTACH SUPPORTING DOCUMENTS TO THE AUDIT COUPON"
      },
      {
         "key": "Baggage",
         "value": "APPLY NORMAL BAGGAGE ALLOWANCE"
      }
   ]
},
{
   "pnrLocator": "FWVOTQ/TWAXGU20170101",
   "dptSequence": "3",
   "bkgStatus": "HK",
   "processId": "0",
   "terms": [
      {
         "key": "Additional Conditions",
         "value": "ATTACH SUPPORTING DOCUMENTS TO THE AUDIT COUPON"
      },
      {
         "key": "Baggage",
         "value": "APPLY NORMAL BAGGAGE ALLOWANCE"
      }
   ]
},
{
   "pnrLocator": "EELSTL/REKNLO20170101",
   "dptSequence": "4",
   "bkgStatus": "HK",
   "processId": "0",
   "terms": [
      {
         "key": "Additional Conditions",
         "value": "ATTACH SUPPORTING DOCUMENTS TO THE AUDIT COUPON"
      },
      {
         "key": "Baggage",
         "value": "APPLY NORMAL BAGGAGE ALLOWANCE"
      }
   ]
},
{
   "pnrLocator": "NSNKRE/ODGRRP20170101",
   "dptSequence": "5",
   "bkgStatus": "HK",
   "processId": "0",
   "terms": [
      {
         "key": "Additional Conditions",
         "value": "ATTACH SUPPORTING DOCUMENTS TO THE AUDIT COUPON"
      },
      {
         "key": "Baggage",
         "value": "APPLY NORMAL BAGGAGE ALLOWANCE"
      }
   ]
},
```

```
            {
               "pnrLocator": "MIWEZH/VZKIGU20170101",
               "dptSequence": "6",
               "bkgStatus": "HK",
               "processId": "0",
               "terms": [
                  {
                     "key": "Additional Conditions",
                     "value": "ATTACH SUPPORTING DOCUMENTS TO THE AUDIT COUPON"
                  },
                  {
                     "key": "Baggage",
                     "value": "APPLY NORMAL BAGGAGE ALLOWANCE"
                  }
               ]
            },
            {
               "pnrLocator": "UEAOPA/RXMRTH20170101",
               "dptSequence": "7",
               "bkgStatus": "HK",
               "processId": "0",
               "terms": [
                  {
                     "key": "Additional Conditions",
                     "value": "ATTACH SUPPORTING DOCUMENTS TO THE AUDIT COUPON"
                  },
                  {
                     "key": "Baggage",
                     "value": "APPLY NORMAL BAGGAGE ALLOWANCE"
                  }
               ]
            }
         ]
      }]
   }
}
```

#### 3.1.3.3.10 Example (Rejection) (XML based rejection)

```
<?xml version=”1.0” ?>
<Reply>
  <REQ>
   <RqstID>1011000530</RqstID>
   <OutstationID>OUT11</OutstationID>
   <RequestDate>20050930</RequestDate>
   <RequestTime>1130</RequestTime>
   <Note>
      <Type>CTC</Type>
      <Value> Boris Walden</Value>
   </Note>
   <Note>
      <Type>RCV</Type>
      <Value> Jurgen Singer </Value>
   </Note>
   <Note>
      <Type>TXT</Type>
      <Value>Not enough Quoted Fare.</Value>
   </Note>
  </REQ>
</Reply>
```

#### 3.1.3.3.11 Example (Rejection) (JSON based rejection)

Mask endpoint: /groupagent/rest/rejectRequest

Exmple: http://localhost:8090/groupagent/rest/rejectRequest

```
{
   "req": {
      "rqstID": "S756371",
      "outstationID": "HOMEOFFIC",
      "requestDate": "20180711",
      "requestTime": "1444",
      "note": [
         {
            "type": "RJT",
            "value": "rejcted with some reason"
         }
      ]
   }
}
```

### *Web Service Interface communication*

### 3.1.4 General Information

Please refer to the PROS web service process specification document for the detail information. G6 expects only one web service provider for this communication.

### 3.1.5 CCITIN

#### *Overview*

The Itinerary Selection interface is one of the interfaces between the PROS Group 6 System and the Reservation System. For each group request which is loaded to the PROS Group 6 System, an itinerary selection request is sent to RES to retrieve all possible flight combinations to accommodate the requested directions. So this interface would require a request message/element and a reply message/element.

The size of the request file is usually below 1 kb and response file size is around 1 to 2 kb per flight option. (Each option constitutes the details for 1 flight for 1 itinerary direction).

The response time for the most complicated requests is expected to be approximately 30-35 seconds.

#### *Message/Signature*

- public String getCcitinOption(String Request) –Request Message

#### *3.1.5.1 Itinerary Selection Request*

##### 3.1.5.1.1 Description

In order to start the evaluation process, the Group System would need to know the flight itineraries that are available in the RES system that match the Group Request. This message will contain the itinerary request information.

Below is the link to the initial XML Schema draft for Request message element and a brief description of the data fields.

#### 3.1.5.1.2 Data Fields

DataSet: Request (version="X.X" airlineVendorID="ZZ") X.X is the current SID version. ZZ is the home base carrier code.

RecordSet: Direction

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Id | NUM 02 | Y | | | | 1 | Reference Number for the direction of travel i.e. 1, 2, 3 etc |
| Origin | CITY | Y | | | | JFK | Directional origin |
| destination | CITY | Y | | | | LHR | Directional destination |
| returnOption | NUM 01 | Y | | | | 0 | 0 = Include all flights that fit within the date and time range (Default)<br>1 = Include preferred flight in all cases, even if it is not in the date and time range, if given.<br>2 = Return just the preferred flight, if given. |
| cityPOS | CITY | N | | | | | PNR Base used. Don't pass this field if there is no value. |
| countryPOS | CHAR 05 | N | | | | DE | Country POS. If not provided, it will be determined using City POS. |
| direction_informative | CHAR 03 | N | | yes/no | | yes | Use yes/no for this field. Marks if direction is informative. |
| origin_informative | CHAR 03 | N | | yes/no | | yes | If the origin is a feeder flight city set this to "yes". |
| destination_informative | CHAR 03 | N | | yes/no | | yes | If the destination is a feeder flight city set this to "yes". |
| via | CITY | N | | | | LAX | Optional city to fly through |
| compartment | CHAR 02 | N | | | | Y | Compartment code |
| OALonly | CHAR 03 | N | | yes/no | | no | If only OAL is allowed, set to "yes" |
| directFlightsOnly | CHAR 03 | N | | yes/no | | no | If the direction should be direct, set to "yes" |
| allowOAL | CHAR 03 | N | | yes/no | | yes | If the direction can include OAL, then set to "yes". |
| pnrLocator | LOCATOR | N | | | | | PNR Base used. Don't pass this field if there is no value. |
| requestId | CHAR 10 | N | | | | A12345 | Unique Reference Number used for identifying a Group Request application.<br><br>This valuable information can be used for debugging purposes by CCITIN interface developer.<br><br>Don't pass this field if there is no value. |
| <DepartureDate> | | Y | | | | | Departure Date information |
| <Flight> | | N | | | | | Flight information. Don't pass this field if there is no value. |

#### *3.1.5.1.2.1 Departure Date Data Fields*

RecordSet: Departure

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| startDate | DATE | Y | | | Y | 20020924 | Start Date |
| endDate | DATE | Y | | | N | 20021024 | End Date |
| frequency | FLAG 07 | Y | | | N | 000100 | The 7 digits are treated as a flag indicator for the day of the week. The first digit represents Monday, and the seventh digit represents Sunday.<br>1011001 = Mon, Wed, Thu, Sun<br>0100110 = Tue, Fri, Sat |
| startTime | TIME 04 | Y | | | N | 0000 | 0000 – 2359 |
| endTime | TIME 04 | Y | | | N | 2359 | 0000 – 2359 |
| orientation | CHAR 02 | Y | | | N | A | A = arrival<br>D = departure (Default) |

#### *3.1.5.1.2.2 Flight Data Fields*

RecordSet: Flight

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| carrier | CARRIER | Y | | | Y | ZZ | Carrier Code |
| flightNumber | FLIGHT 05 | Y | | | N | 00100 | Flight Number |
| sequenceNumber | NUM 02 | Y | | | N | 1 | Flight Sequence Number |
| setNumber | NUM 02 | N | | | | 1 | Flight Set Number |
| orgin | CITY | N | | | N | HOU | Preferred flight origin. Don't pass this field if there is no value. |
| Destination | CITY | N | | | N | LAX | Preferred flight destination. Don't pass this field if there is no value. |
| Date | DATE | N | | | N | 20051024 | If no departure date is specified, field must be blank. Don't pass this field if there is no value. |

### 3.1.5.1.3 Processing Rules

### 3.1.5.1.4 Loader Behavior

### 3.1.5.1.5 Schema

The content of schema can be found at /config/interface/xsd/ccitinreq.xsd file.

### 3.1.5.1.6 Example

```
<?xml version="1.0"?>
<Request version="1.4" airlineVendorID="ZZ">
  <Direction id="1"
    origin="HOU"
    destination="LAX"
    returnOption="0"
```

```
   cityPOS="HOU"
   pnrLocator="B32DT20050812"
   requestId="A12345">
  <DepartureDate startDate="20020924"
   endDate="20021024"
   frequency="0001000"
   startTime="0000"
   endTime="2359"
   orientation="A"/>
  <Flight carrier="ZZ"
   flightNumber="00100"
   sequenceNumber="1"
   setNumber="1"
   orgin="HOU"
   destination="LAX"
   date="20021024"/>
  </Direction>
</Request>
```

### *3.1.5.2 Itinerary Selection Response*

#### 3.1.5.2.1 Description

This will contain the response from the RES system with the set of possible Itineraries that would match the request. The information that must be passed to the Group System is for the Traveled Legs and for ALL Segments of the applicable flights.

#### 3.1.5.2.2 Data Fields

DataSet: Response
RecordSet: Direction

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| id | NUM 02 | Y | | | | 1 | The direction starts from 1. It must match the request directional information. |
| errMsg | CHAR 50 | N | | | | "No option available" | If there is any error, provide the error message. |
| <Option> | | N | | | | | Option information |

##### *3.1.5.2.2.1 Option Data Fields*

RecordSet: Option

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| id | NUM 02 | Y | | | | 1 | The option number must start from 1 within each direction. |
| EFT | TIME 04 | N | | | | 0130 | The elapsed flying time of the option. |
| <Flight> | | | | | | | Flight information |

##### *3.1.5.2.2.2 Flight Data Fields*

RecordSet: Flight

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| carrier | CARRIER | Y | | | | ZZ | The option number must start from 1 within each direction. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| flightNumber | FLIGHT 05 | Y | | | | 00001 | Flight Number |
| marketingCarrier | CARRIER | N | | | | ZZ | Marketing Carrier Code |
| marketingFlightNumber | FLIGHT 05 | N | | | | 00001 | Marketing Flight Number |
| sequenceNumber | NUM 02 | Y | | | | 1 | Flight Sequence Number. This number starts from 1. |
| departureDate | DATE | Y | | | | 20031005 | This Departure Date must correspond with the one of the first leg of the flight (Flight Departure Date), regardless of the actual departure date of the leg/segment traveled by the group. |
| Origin | CITY | Y | | | | HOU | Group boarding point. |
| Destination | CITY | Y | | | | LAX | Group off-boarding point. |
| departureTime | TIME 04 | Y | | | | 0000 | Departure Time |
| arrivalTime | TIME 04 | Y | | | | 0000 | Arrival Time |
| noneControlFlag | FLAG 01 | N | | | | 1 | This flag indicates the flight does not exist in the OD/P. By default, do not send this flight if the flight is available in OD/P. |
| arrivalDate | Date | N | | | | 20110101 | Group off-boarding arrival date. |
| aircraftType | CHAR 04 | N | | | | 321 | Aircraft type code |
| EFT | TIME 04 | N | | | | 0130 | Elapsed flying time of the flight. |
| <Leg> | | Y | | | | | Leg information |
|  | | Y | | | | | Segment information |

#### *3.1.5.2.2.3 Leg Data Fields*

This record must contain information **only** for traveled legs.

RecordSet: Leg

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| dateOffset | NUM 02 | Y | | | | 0 | The date offset value must be the difference between the group travel date and the flight departure date for this leg. For example we have a flight with two legs. First leg of flight travels on Jan 25, and second leg travels on Jan 26. If the travel O&D is on the second leg of this flight, the date offset is 1 for this leg. |
| destination | CITY | Y | | | | LAX | Leg Destination |
| sequenceNumber | NUM 02 | Y | | | | 1 | Leg Sequence Number. Segment must start with 1 |
| aircraftConfig | CHAR 06 | N | | | | | Aircraft Configuration Key To be use for Convertible Seat logic **only**. Don't pass this field if there is no value. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| allowBooking | FLAG 01 | Y | | | | 1 | 1 = Booking allowed (Default)<br>0 = No Booking Allowed |
| allowSale | FLAG 01 | Y | | | | 1 | 1= Sale allowed (Default)<br>0 = No sale allowed |
| allowGroup | FLAG 01 | Y | | | | 1 | 1= Group Allowed (Default)<br>0 = No Group Allowed |
| allowWaitList | FLAG 01 | Y | | | | 1 | 1 = Waitlist allowed (Default)<br>0 = No waitlist allowed |
| <Compartment> | | Y | | | | | Leg compartment |
| <Arcf> | | N | | | | | Aircraft Configuration List<br>This field should only be provided when GRMS is used without a connection to ODWI and obtains revenue management data from the reservation system.<br><br>If this field is used, the Aircraft Confguration Key should not be provided on the Leg level. Instead, it should be provided with the Aircraft Confguration List. |
| <InvMap> | | N | | | | | Inventory Map<br>This field should only be provided when GRMS is used without a connection to ODWI and obtains revenue management data from the reservation system. |

#### *3.1.5.2.2.4 Aircraft Configuration List*

RecordSet: Arcf

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| key | ASCII | Y | | | | CL6 | Aircraft Configuration key |
| <Configuration> | | Y | | | | | |

### 3.1.5.2.2.4.1 Aircraft Configuration Data Fields

RecordSet: Configuration

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| id | ASCII | Y | | | | 003 | Configuration ID |
| <Cmp> | | Y | | | | | |

#### *3.1.5.2.2.4.1.1 Aircraft Configuration Compartment Data Fields*

RecordSet: Cmp

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| code | SYMBOL 02 | Y | | | | C | Compartment code |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| phyCap | NUMBER 03 | Y | | | | 120 | Physical Capacity |

#### *3.1.5.2.2.5 Inventory Map*

RecordSet: InvMap

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| key | ASCII | Y | | | | 902 | Inventory Map key |
| <Compartment> | | Y | | | | | |

#### 3.1.5.2.2.5.1 Inventory Map Compartment Data Fields

RecordSet: Compartment

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| code | SYMBOL 02 | Y | | | | C | Compartment code |
| <Class> | | Y | | | | | |

##### *3.1.5.2.2.5.1.1 Inventory Map Class Data Fields*

RecordSet: Class

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| code | SYMBOL 02 | Y | | | | M | Class code |
| parentClass | SYMBOL 02 | Y | | | | B | Parent Class |

#### *3.1.5.2.2.6 Leg Compartment Data Fields*

RecordSet: Compartment

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| code | SYMBOL 02 | Y | | | | F | Compartment Code. |
| capacity | NUM 03 | Y | | | | 123 | Physical Capacity |
| adjustedCapacity | NUM 03 | N | | | | 123 | Adjusted Capacity<br>This field should only be provided when GRMS is used without a connection to ODWI and obtains revenue management data from the reservation system. |
| totalBooked | NUM 03 | Y | | | | 100 | Total Compartment Booked |
| odFactor | NUM 03 | Y | | | | 100 | This is the factor used to sum the marginal fares to form a directional marginal fare in the case of multiple legs that form a direction. This is used if the OD is selected as the OD factor.<br>Default: 100. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| oalAvailability | NUM 03 | N | | | | 100 | OAL availability passing from the Reservation system. |
| groupBooked | NUM 03 | N | | | | 20 | Compartment Group Bookings. If this field is provided, then the aggregration of the Class level groupBooked will be ignored. Most Segment-control carriers do not have the Leg Class information. Thus, this field can provide the missing information to help on the AP rule setting. |
| groupCapacityPercent | NUM 02 | N | | | | 50 | Percentage of capacity that groups may use |
| totalAU | NUM 03 | N | | | | 22 | Total AU<br>This field should only be provided when GRMS is used without a connection to ODWI and obtains revenue management data from the reservation system. |
| fcstOut | NUM 03 | N | | | | 5 | Forecasted Out<br>This field should only be provided when GRMS is used without a connection to ODWI and obtains revenue management data from the reservation system. |
| fcstDmdLF | FLT 11.5 | N | | | | 89.6 | Forecasted Demand Load Factor<br>This field should only be provided when GRMS is used without a connection to ODWI and obtains revenue management data from the reservation system. |
| <Bid Price> | | N | | | | | Bid Price vector<br>This field should only be provided when GRMS is used without a connection to ODWI and obtains revenue management data from the reservation system. |
| <Class> | | | | | | | Leg Class. It is required for the Leg control carriers |

#### *3.1.5.2.2.7 Leg Bid Price Data Fields*

RecordSet: Bid Price

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| vector | ASCII | Y | | | | "100,95,80,50,25,1,0" | Bid Price Vector – List of bid prices, comma separated, ordered from 0 seats remaining to X seats remaining |
| index | NUM 03 | N | | | | 17 | Index of current seat in the bid price vector. If index is not provided, it is calculated as Leg Compartment AU – Leg Compartment Booked. |

#### *3.1.5.2.2.8 Leg Class Data Fields*

RecordSet: Class

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| code | SYMBOL 02 | Y | | | | C | Class Code |
| parentCode | SYMBOL 02 | Y | | | | C | Parent Class code |
| nestedAU | NUM 03 | Y | | | | 100 | Class Nested AU |
| totalBooked | NUM 03 | Y | | | | 100 | Class Total Bookings |
| groupBooked | NUM 03 | Y | | | | 20 | Class Group Bookings |
| allowBooking | FLAG 01 | Y | | | | 1 | 1 = Booking allowed (Default)<br>0 = No Booking Allowed |
| allowSale | FLAG 01 | Y | | | | 1 | 1= Sale allowed (Default)<br>0 = No sale allowed |
| allowGroup | FLAG 01 | Y | | | | 1 | 1= Group Allowed (Default)<br>0 = No Group Allowed |
| allowWaitList | FLAG 01 | Y | | | | 1 | 1 = Waitlist allowed (Default)<br>0 = No waitlist allowed |
| seatsBlocked | NUM 03 | Y | | | | 100 | Seats Block (Convertible seat only)<br>This field can be used to set a minimum number of seats in a certain class that cannot be converted.<br>Default to 0 |

#### *3.1.5.2.2.9 Segment Data Fields*

This record contains information for **all** segments of this flight. It may contain the segment not satisfy the directional O&D.

RecordSet: Segment

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| origin | CITY | Y | | | | HOU | Segment origin |
| destination | CITY | Y | | | | LAX | Segment destination |
| sequenceNumber | NUM 02 | Y | | | | 1 | Segment Sequence Number.<br>Segment must start with 1 |
| allowBooking | FLAG 01 | Y | | | | 1 | 1 = Booking allowed (Default)<br>0 = No Booking Allowed |
| allowSale | FLAG 01 | Y | | | | 1 | 1= Sale allowed (Default)<br>0 = No sale allowed |
| allowGroup | FLAG 01 | Y | | | | 1 | 1= Group Allowed (Default)<br>0 = No Group Allowed |
| allowWaitList | FLAG 01 | Y | | | | 1 | 1 = Waitlist allowed (Default)<br>0 = No waitlist allowed |
| <Compartment> | | Y | | | | | Segment compartment |

#### *3.1.5.2.2.10 Segment Compartment Data Fields*

RecordSet: Compartment

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| code | SYMBOL 02 | Y | | | | F | Compartment Code. |
| groupBooked | NUM 03 | Y | | | | 123 | Compartment Group Bookings |
| allowAvailability | FLAG 01 | N* | | | | 1 | This is an optional field.<br>This field can be ignored by customers who provide NestedAU in the segment class level.<br>The purpose of the field is to support segment based customers who can only provide nestedAvailability instead of nestedAU.<br>If this field is set to 1, it will tell the system (G) to use the nestedAvailability field (instead of nestedAU) for all the availability calculations.<br>Note: If the flag is set to 1, and nestedAvailabilty field is not populated, G6 will treat the nestedAvailability value as 0. |
| <Class> | | Y | | | | | Compartment Class |

#### *3.1.5.2.2.11 Segment Class Data Fields*

RecordSet: Class

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| code | SYMBOL 02 | Y | | | | C | Class Code |
| parentCode | SYMBOL 02 | N | | | | C | Parent Class code. It is required for the segment control carriers. Don't pass this field if there is no value. |
| nestedAU | NUM 03 | N* | | | | 100 | **Segment Control Carriers**:<br>Class segment Nested AU.<br>This field is required.<br><br>**Leg Control Carriers**:<br>This filed is optional for leg control carriers.<br>Don't pass this value if there is no value. |
| limit | NUM 03 | N* | | | | 100 | **Segment Control Carriers**:<br>This field is optional for the Segment control carreiers.<br><br>**Leg Control Carriers**:<br>Class segment limit.<br>This field is required. If not known, pass 999. |
| totalBooked | NUM 03 | Y | | | | 100 | Class Total Bookings |
| groupBooked | NUM 03 | Y | | | | 20 | Class Group Bookings |
| classPos | CHAR 10 | N | | | | All | Class point of sales. It is required for the segment control carriers. Don't pass this field if there is no value. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| allowBooking | FLAG 01 | Y | | | | 1 | 1 = Booking allowed (Default)<br>0 = No Booking Allowed |
| allowSale | FLAG 01 | Y | | | | 1 | 1= Sale allowed (Default)<br>0 = No sale allowed |
| allowGroup | FLAG 01 | Y | | | | 1 | 1= Group Allowed (Default)<br>0 = No Group Allowed |
| allowWaitList | FLAG 01 | Y | | | | 1 | 1 = Waitlist allowed (Default)<br>0 = No waitlist allowed |
| nestedAvailability | NUM 03 | N* | | | | 999 | This field is recommended to be used only when nestedAU is not available from RES.<br>**Segment Control Carriers**:<br>Class segment Nested availability. This field is optional. . It is designed for any segment carrier that is not able to provide NestedAU value.<br>Don't pass this value if there is no value.<br><br>**Leg Control Carriers**:<br>This filed is optional for leg control carriers.<br>Don't pass this value if there is no value. |
| odAvailability | NUM 03 | N | | | | 9 | This field is optional. It should only be sent if you intend to use class availability during evaluation. |

#### 3.1.5.2.3 Processing Rules

None

#### 3.1.5.2.4 Loader Behavior

Insert/Update

#### 3.1.5.2.5 Schema

The content of schema can be found at /config/interface/xsd/ccitinrep.xsd file.

#### 3.1.5.2.6 Example

```
<?xml version="1.0"?>
<Response>
  <Direction id="1">
   <Option id ="1">
   <Flight carrier="ZZ"
      flightNumber="00001"
      sequenceNumber="1"
      departureDate="20031005"
      origin="HOU"
      destination="LAX"
      departureTime="0000"
      arrivalTime="0000">
     <Leg dateOffset="0"
      destination="LAX"
      sequenceNumber="1"
      aircraftConfig="ZZ0001"
      allowBooking="1"
```

```
      allowSale=”1”
      allowGroup=”1”
      allowWaitList=”1”>
        <Compartment code=”Y”
        capacity=”123”
        totalBooked=”100”
        odFactor=”100”>
        <Class code=”Y”            // required for the leg control carriers
        parentCode=”Y”
        nestedAU=”100”
        totalBooked=”100”
        groupBooked=”20”
        allowBooking=”1”
        allowSale=”1”
        allowGroup=”1”
        allowWaitList=”1”
        seatsBlocked=”100”/>
        </Compartment>
     </Leg>
   <Segment origin=”HOU”
      destination=”LAX”
      sequenceNumber=”1”
      allowBooking=”1”
      allowSale=”1”
      allowGroup=”1”
      allowWaitList=”1”>
      <Compartment code=”Y”
       groupBooked=”100”>
        <Class code=”Y”
        parentCode=”Y”      // required for the segment control carriers
        nestedAU=”100”      // required for the CITP additional field
        limit=”100”
        totalBooked=”100”
        groupBooked=”20”
        classPos=”ALL”     // required for the segment control carriers
        allowBooking=”1”
        allowSale=”1”
        allowGroup=”1”
        allowWaitList=”1”/>
      </Compartment>
  </Segment>
</Flight>
</Option>
</Direction>
</Response>
```

**text error messages** must be provided only when errors occur.

```
<?xml version=”1.0”?>
<Response>
<Direction id=”1” errMsg=”Communication Failed”>
</Direction>
</Response>
```

**Note**: Group expects the classes to be passed in the descending nested order. Parent classes appear before the child classes.

For example,

```
<Compartment code = "Y"
   <Class code="Y"
   parentCode="Y"
                              …..>
   <Class code="H"
   parentCode="Y"
                              …..>
   <Class code="K"
   parentCode="H"
                              …..>
   <Class code="B"
   parentCode="K"
                              …..>

   <Class code="X"   // second cluster if any
   parentCode="Y"
                              …..>
   <Class code="T"
   parentCode="X"
                              .....>

                              ……..
  </Compartment>
```

The nested classes must be clustered together in the descending order (parent class display first). In a hybrid nesting case, a second cluster group must follow the first cluster group.

### 3.1.6 PNRGen

#### *Overview*

After the Group Request is evaluated from within the Group System and the appropriate itineraries selected for the Group, the information is then communicated to the RES system. The RES system would reply back with a confirmation that the PNR data in RES has been updated.

#### *Message/Signature*

- public String getPNRResult (String Request) –Request Message

#### *3.1.6.1 PNRGen Request*

##### 3.1.6.1.1 Description

This message will contain the PnrGen Request information that will be passed from the Group System to the RES.

##### 3.1.6.1.2 Data Fields

DataSet: Request (version="X.X" airlineVendorID="ZZ") X.X is the current SID version. ZZ is the home base carrier code.

This group record contains general Group information. There can only be one Group record for every group shell to be created/updated.

RecordSet: Group

**Attributes**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| id | NUM 10 | Y | | | | 123 | Unique identifier in the G6 system assigned to each request.<br>For the PNR-based system, the id will be 0 when performing a request rejection. |
| seriesId | NUM 10 | N | | | | 100 | Ad hoc requests will have no value. Don't pass this field if there is no value. |
| userId | CHAR 128 | Y | | | | JOHN | User ID |
| outstation | CHAR 09 | Y | | | | HOU | Outstation ID |
| groupName | CHAR 50 | Y | | | | Leisure | Group Name |
| customerType | NUM 10 | Y | | | | 123 | Customer Type |
| groupType | CHAR 02 | Y | | | | A | A = ad hoc (Default)<br>S = series |
| mktOrigin | CITY | N | | | | HOU | Default to first direction of origin. Don't pass this field if there is no value. |
| mktDestination | CITY | N | | | | LAX | Default to first direction of destination. Don't pass this field if there is no value. |
| cityPos | CITY | N | | | | HOU | City POS |
| districtPos | CITY | N | | | | US | District POS |
| iata | CHAR 10 | N | | | | 12345 | Agency IATA. Don't pass this field if there is no value. |
| officeCode | CHAR 10 | N | | | | 12345 | Office Code. Don't pass this field if there is no value. |
| agencyName | CHAR 50 | N | | | | World Travel | Agency Name. Don't pass this field if there is no value. |
| realFare | NUM 10 | N | | | | 300 | Real Fare for the group shell. Weighted average value for this group shell. This field is for the referenc only. The actual fare information must be the real fare under PNR level. Please check 3.5.3.1.2.1 |
| commPercent | NUM 02 | N | | | | 5 | Commission Percentage |
| passes | NUM 03 | N | | | | 1 | Passes. Don't pass this field if there is no value. |
| requestId | CHAR 10 | N | | | | R12345 | Request ID. Don't pass this field if there is no value. |
| totalTripFare | NUM 10 | N | | | | 400 | Don't pass this field if there is no value. |
| keyAccount | CHAR 06 | N | | | | IAH | keyAccount field associated with the iata. This field is retrieved from Agency table using the iata as a reference value.<br><br>This field will only be populated if the KeyAccount value is populated in the Agency table. |

**Elements**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| <PNR> | | Y | | | | | PNR information |
| <PRO> | | N | | | | | Text information. Don't pass this field if there is no value. |
| <PRM> | | N | | | | | Miscellaneous Information. Don't pass this field if there is no value. |

#### *3.1.6.1.2.1 PNR Data Fields*

This PNR record contains PNR header information.

RecordSet:PNR

**Attributes**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| index | NUM 03 | Y | | | | 1 | Unique identifier in the G6 system assigned to each request |
| locator | LOCATOR | N | | | | | Includes Creation Date<br>This field will only be filled out in case of re-evaluation and PNR request. Don't pass this field if there is no value. |
| primIndicator | FLAG 01 | Y | | | | 1 | 0 = PNR child<br>1 = PNR parent (Default) |
| utilRate | NUM 03 | N | | | | | GHI feature. Materialization rate assigned to the PNR by the Materialization process in G6. |
| numberInParty | NUM 03 | N | | | | | GHI feature. Original seats requests by the group. |
| realFare | NUM 10 | N | | | | 300 | Real Fare for the PNR. |
| localFare | NUM 10 | N | | | | 300 | Local Fare for the PNR. |
| ancillaryRevenue | NUM 10 | N | | | | 150 | This is the sum of all ancillary revenue for the selected options. |
| adjustedRealFare | NUM 10 | N | | | | 300 | Adjusted Net Real Fare for the PNR. It does not contain the Ancillary Revenue. |

**Elements**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| <Itinerary> | | Y | | | | HOU | Itinerary information |
| <Deals> | | N | | | | | Deal message |

#### *3.1.6.1.2.2 Text Data Fields*

This PRO record contains remarks that will be included in the PNR upon generation. There can be multiple PRO records per PNR block. In general, the value is the text message for

the PRO. A special type of reminder is available for the feature of name firming. The name firming format will look like the following.

```
<PRO  noteType="NFM"
            value="1 90 20070101
                   2 60 20070201
                   3 30 20070301
                   4 0 20070401
                   5 0 20070401"/>
```

RecordSet: PRO

**Attributes**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| noteType | CHAR 04 | Y | | | | PRC | Predefined four letter identifier (i.e. SSR, PRC,… etc) |
| value | CHAR 255 | Y | | | | CONTACT INFORMATION | Includes Creation Date<br>This field will only be filled out in case of re-evaluation and PNR request. |

#### *3.1.6.1.2.3 Miscellaneous Data Fields*

This PRM record is a PNR miscellaneous record. It contains one PRM record per Group.

RecordSet: PRM

**Attributes**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| programName | CHAR 50 | N | | | | program | Program Name. Don't pass this field if there is no value. |
| tourName | CHAR 50 | N | | | | tourName | Tour Name. Don't pass this field if there is no value. |
| currencyCode | CHAR 03 | N | | | | currencyCode | Local Currency Code. Don't pass this field if there is no value. |
| localFare | NUM 10 | N | | | | localFare | Local Fare. Don't pass this field if there is no value. |
| refPnrLocator | LOCATOR | N | | | | refPnrLocator | Reference PNR Locator. Don't pass this field if there is no value. |
| exchangeRate | FLT 11.5 | N | | | | 1.5 | Rate-of-Exchange. Don't pass this field if there is no value. |
| SourceCRS | CHAR 05 | N | | | | AM | Central Reservation System. This field will not be passed if there is no value. |

#### *3.1.6.1.2.4 Itinerary Data Fields*

This Itineary record contains PNR/Segment level information. There can be multiple Itinerary records for each PNR record.

RecordSet: Itinerary

**Attributes**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| lineNum | NUM 03 | Y | | | | 1 | Itinerary Line Number |
| carrier | CARRIER | Y | | | | ZZ | Includes Creation Date<br>This field will only be filled out in case of re-evaluation and PNR request. |
| flightNumber | FLIGHT 05 | Y | | | | 00001 | Flight Number |
| departureDate | DATE | Y | | | | 20031005 | Format: YYYYMMDD |
| origin | CITY | Y | | | | HOU | Segment Origin |
| destination | CITY | Y | | | | LAX | Segment Destination |
| classCode | SYMBOL 02 | Y | | | | C | Class Code |
| paxNumber | NUM 03 | Y | | | | 20 | Number Passenger Booked |
| actionCode | SYMBOL 02 | N | | | | KK | Action requires status code |
| classPos | CHAR 10 | N | | | | | Class POS |

#### *3.1.6.1.2.5 Deal Data Fields*

#### *3.1.6.1.2.6*

The data fields contain necessary information to support the Deal Code feature.

RecordSet: Deals

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| Deal | CHAR 1024 | Y | | | | Some Value | Each Deal Element will have one of these entries. |

**Deal Attributes**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| key | CHAR 32 | Y | | | | Some Key | Key for a Deal |

### 3.1.6.1.3 Processing Rules

None

### 3.1.6.1.4 Loader Behavior

Insert/Update

### 3.1.6.1.5 Schema

The content of schema can be found at /config/interface/xsd/pnrgenreq.xsd file.

### 3.1.6.1.6 Example

```
<?xml version="1.0"?>
<Request version="1.4" airlineVendorID="ZZ">
  <Group id="1"
```

```
   seriesId="1"
   userId="LEE"
   outstation="HOU"
   groupName="Leisure"
   customerType="123"
   groupType="A"
   mktOrigin="HOU"
   mktDestination="LAX"
   cityPos="HOU"
   districtPos="US"
   iata="12345"
   officeCode="officeCode"
   agencyName="World Travel"
   realFare="100"
   commPercent="5"
   passes="12"
   requestId="R12345"
   totalTripFare="100">
   <PNR index="1"
      locator="Locator"
      primIndicator="1"
      realFare="100"
      localFare="100"
      >
      <Itinerary lineNum="1"
         carrier="ZZ"
         flightNumber="00001"
         departureDate="20031005"
         origin="HOU"
         destination="LAX"
         classCode="C"
         paxNumber="20"
         actionCode="KK"
         classPos="all"/>
      <Deals>
        <Deal key="Key1">Value1</Deal>
        <Deal key="Key2">Value2</Deal>
      </Deals>
  </PNR>
  <PRO  noteType="SSR"
     value="SSR value"/>
  <PRO  noteType="OSI"
     value="OSI value"/>
  <PRO  noteType="IMP"
     value="IMP value"/>
  <PRO  noteType="HLD"
     value="HLD value"/>
  <PRO noteType="PRC"
     value="Contact Info"/>
  <PRM programName="program"
   tourName="tourName"
   currencyCode="USD"
   localFare="100"
   refPnrLocator="A12345"
   exchangeRate="1.0"/>
  </Group>
</Request>
```

### *3.1.6.2 PNRGen Response*

#### 3.1.6.2.1 Description

This message will be the reply from the RES in response to the PnrGen request. The reply will contain the Pnr Locator or Pnr Reference Number.

#### 3.1.6.2.2 Data Fields

It contains general Group/PNR information. There must be only one Group record for each Group Shell.

DataSet: Response
RecordSet: Group

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| id | NUM 10 | Y | | | | C | Unique identifier in the G6 system assigned to each request |
| errMsg | CHAR50 | N | | | | "Communication error" | If there is any error, provide the error message. |
| <PNR> | | N | | | | | PNR information |

##### *3.1.6.2.2.1 PNR data fields*

It contains PNR header information. There can be multiple PRR records;

RecordSet: PNR

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| index | NUM 03 | Y | | | | 1 | Unique identifier in the G6 system assigned to each request |
| locator | LOCATOR | Y | | | | | PNRLOCATOR value return from the reservation system. |
| errMsg | CHAR 50 | N | | | | | If there is any error, provide the error message. |
| rejectFlag | NUM 01 | N | | | | 0 | If GRMS has to automatically reject the request after this attempted PNRGEN then set this flag to 1 and set the errMsg to why GRMS needs to reject. |
| <Itinerary> | | N | | | | | Itinerary information. |

##### *3.1.6.2.2.2 Itinerary data fields*

It contains PNR/Segment information. There can be multiple Itinerary records for each PNR record.

RecordSet: Itinerary

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| lineNum | NUM 03 | Y | | | | 1 | Itinerary Line Number |
| status | CHAR 02 | Y | | | | HK | Booking status |

#### 3.1.6.2.3 Processing Rules

#### 3.1.6.2.4 Loader Behavior

#### 3.1.6.2.5 Schema

The content of schema can be found at /config/interface/xsd/pnrgenrep.xsd file.

#### 3.1.6.2.6 Example

```
<?xml version=”1.0”?>
<Response>
   <Group id =”1”>
   <PNR index=”1” locator=”AAAAA20051010”>
     <Itinerary lineNum=”1” status=”HK”/>
   </PNR>
   </Group>
</Response>
```

### 3.1.7 PNR Refresh

#### *Overview*

After the Group Request is loaded from within the Group System, the same PNR information in RES system may be updated. This PNR Refresh interface will retrieve the latest PNR imformation form RES system and load the latest PNR information to the G6.

#### *Message/Signature*

- public String getPNRImage(String Request) –Request Message

#### *3.1.7.1 PNR Refresh Request*

#### 3.1.7.1.1 Description

This message will contain the Pnr Request information that will be passed from the Group System to the RES.

#### 3.1.7.1.2 Data Fields

DataSet: Request (version=”X.X” airlineVendorID=”ZZ”) X.X is the current SID version. ZZ is the home base carrier code.

This PNR record contains general PNR information. There must be one PNR record for every PNR request to be updated.

RecordSet: PNR

**Attributes**

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| pnrLocator | LOCATOR | Y | | | | B32DT20050812 | Unique Identifier for current and historical PNRs in the G6 system.<br>In case of add-on group requests, the PNR Locator is empty for the mask input. |

| Data Field | Format | Req'd | Referential | Validity | PK | Example | Description / Note |
|---|---|---|---|---|---|---|---|
| primPnrLocator | LOCATOR | N | | | | B32DT2005 0812 | Unique Identifier for current and historical PNRs in the G6 system.<br>In case of split PNRs, the locator of the Parent (or Master) PNR must go into this field. This field will decide what primary PNR indicator to use during the PNRGen operation.<br>In case of add-on group requests the Primary PNR Locator is empty. |
| refPnrLocator | LOCATOR | N | | | | B32DT2005 0812 | In case of add-on group requests the Reference PNR is the PNR that is being added to.<br>In the case of a request originated from a GDS PNR, the Reference PNR will be the GDS PNR. Don't pass this field if there is no value. |
| requestId | CHAR 10 | Y | | | | Group433 | Unique Reference Number used for identifying a Group Request application. |
| outstation | CHAR 09 | Y | | | | OUT11 | Airlines Outstation/Sales office where request originates from. |
| requestDate | DATE | Y | | | | 20050930 | Date group booking request submitted to G6.<br>Format: YYYYMMDD |
| requestTime | TIME 04 | Y | | | | 1130 | Time group booking request submitted to G6.<br>Format: 0000 – 2359 |
| requestTypeFlag | FLAG | Y | | | | 0 | Flag to identify different "types" of requests:<br>0 = "Regular" request<br>1 = Add-on request – to increase the number of seats originally requested<br>2 = Re-evaluation request – to modify and already booked group. |

#### 3.1.7.1.3 Processing Rules

None

#### 3.1.7.1.4 Loader Behavior

Insert/Update

#### 3.1.7.1.5 Schema

The content of schema can be found at /config/interface/xsd/refreshreq.xsd file.

#### 3.1.7.1.6 Example

```
<?xml version="1.0"?>
<Request version="1.4" airlineVendorID="ZZ">
  <PNR pnrLocator="AAAAAA20121231"
   requestId ="AAAAAA2012"
   outstation="HOU"
```

```
   requestDate="20120101"
   requestTime ="1130"
   requestTypeFlag ="0"/>
</Request>
```

### *3.1.7.2 PNR Refresh Response*

#### 3.1.7.2.1 Description

This message will be the reply from the RES in response to the PNR Refresh request. The reply will contain the Ah Hoc request file format.

#### 3.1.7.2.2 Data Fields

The PNR Refresh data format must be identical to the one in the 3.3.2.1 Ad Hoc Request.

## *Reservation migration*

### 3.1.8 General Information

When a GRMS customer changes RES Systems, it is necessary to make some changes in the GRMS data. Specifically, changes related to active PNRs record locators as these will certainly change with the new RES.

The purpose of this document is to define the format required to migrate the active record locators in GRMS (related to the old RES system) to the new locators of the new RES.

### 3.1.9 Input Format

In order to execute this conversion, the new RES vendor needs to provide the "Migration PNR Reference file". This is a text file that will list in each row the new PNRs locator and its creation date, along with the old PNR locator and its creation date. These 4 elements need to be comma separated. With this, the format will be as follows:

| Element Position | Element Name | MAX Width | Mandatory | Comment |
|---|---|---|---|---|
| 1 | New PNR Locator | 6 | yes | Record locator of the PNR from the *new* RES system |
| 2 | comma | 1 | yes | Separator. Set to " , " |
| 3 | New PNR Creation Date | 8 | yes | Creation date of the PNR in the *new* RES system. Must be in YYYYDDMM format |
| 4 | comma | 1 | yes | Separator. Set to " , " |
| 5 | Old PNR Locator | 6 | yes | Record locator of the PNR from the *old* RES system. This is in fact the locator that must exist in GRMS before the conversion. |
| 6 | comma | 1 | yes | Separator. Set to " , " |
| 7 | Old PNR Creation Date | 8 | no | Creation date of the PNR in the *old* RES system. Must be in YYYYDDMM format.<br><br>Note: This field can be left empty if the old PNR locator's creation date cannot be provided.<br>If this field is left empty the script will use the newest creation date of the matching PNR locator that exists in GRMS DB. If more than one PNR locator with the same locator string then only the newest one will be converted. |

**Note:** There shall be 1 line per PNR locator, no blank elements, and no empty lines.

### 3.1.10 Example

```
QZ45IR, 20081231, 2NR9V, 20080617
QZ45IW, 20081231, R1F8C,
QZ45IX, 20081231, R4FSD,
QZ45IY, 20081231, KM7N1,
QZ45IZ, 20081231, S45RW, 20080228
…
```

Note: Old PNR Creation date of the 3 middle lines (locators **QZ45IW, QZ45IX** and **QZ45IY**) is intentionally left out to show the optional nature of this field.

### 3.1.11 File

| Load Order | Data | Pattern | Task Name |
|---|---|---|---|
| 1 | Reservation Migration | **PNR_LOC_INPUT.txt** | NA |

# 4. Interface Communication Protocol

### *Overview*

This section will serve as the requirements for the file transfer protocol. The protocol will be used to load files into the Group System. The interface is based on a polling mechanism. To accomplish this, the system observes a specified directory, looking for files that match certain patterns (called "*file patterns*"). The system determines which file pattern to use based on a task and possibly a task order. Once it finds the file matching the appropriate file pattern, it loads the file immediately. This process is described in more detail, below. Other files, not included in the ordered task, can be loaded at any time other than the time the ordered task is running.

### *File Loading Tasks*

#### 4.1.1 Loading Files

Each file type has a specific task that looks for a particular file pattern. When a new file has been copied to the proper location and is ready to be loaded, the task is started manually using a "runTask" script. Listed below are files considered to be "On-Demand" or Optional, along with the relevant task name and file patterns required to load the file. Keep in mind the proper load order for these files when setting up a new system. For more information, see I&M Guide.

| Load Order | Data | Pattern | Task Name |
|---|---|---|---|
| 1 | Travel Agency | .AGENCY. | pLoadAgency |
| 2 | City Point of Sale | .CITYPOS. | pLoadCity |
| 3 | District Point of Sale | .DISTPOS. | pLoadDist |
| 4 | Market O&D | .MKTOD. | pLoadMktOD |
| 5 | Market Origin | .MKORG. | pLoadMarketOrigin |
| 6 | Market Fare | .MKFARE. | pLoadMarketFare |
| 7 | Market Special fare | .MKSPFARE. | pLoadMarketSpecialFare |
| 8 | Market Matrix | .MKTMTRIX. | pLoadMarketMatrix |
| 9 | Market Matrix Date Range | .MKTMATRIXDATERANGE. | pLoadMarketmatrixDateRange |
| 10 | Fare Matrix | .MKTFAREADJUSTMENT. | pLoadMarketFareAdjustment |
| 11 | Customer Type | .CUSTTYPE. | pLoadCustomerType |
| 12 | Exchange Rate | .EXCHANGE. | pLoadExchangeRate |
| 13 | Peak/Off-peak Period | .POP. | pLoadPOP |
| 14 | Traffic Area | .TRAFFIC. | pLoadTraffic |
| 15 | Region | .REGION. | pLoadRegion |
| 16 | Country | .COUNTRY. | pLoadCountry |
| 17 | OAL Fare | .OALFARE. | pLoadOALFare |
| 18 | Marketing Flight | .MKTFLTFARE. | pLoadMarketingFlightFare |

| Load Order | Data | Pattern | Task Name |
|---|---|---|---|
| 19 | Deposit Deadline | .DEPOSITDEADLINE. | pLoadDepositDeadline |
| 20 | Name Firming Deadline | .NAMEFIRMINGDEADLINE. | pLoadNameFirmingDeadline |
| 21 | Flight Group Capacity Control | .FLIGHTGROUPCAPACITY. | pLoadFlightGroupCapacity |

Once the task is triggered, a single file matching the file pattern will be loaded from a specified directory. Once the file is loaded, it will be renamed to indicate the load is complete. If multiple files are required, then the task will have to be triggered once for each file.

### 4.1.2 Transactional Files

For the "Transactional" files, a single task will be configured with a specified number of each file and a file order. The following represents the Transactional files, file order, file patterns, and number of each file:

| Load Order | Data | Pattern | Individual Load Task |
|---|---|---|---|
| 1 | Request Load | .REQUEST. | pLoadRequest |
| 2 | PNRload | .PNR. | pLoadPnr |
| 3 | AcceptLoad | .ACCPT. | pLoadAcceptance |
| 4 | SeriesLoad | .SERIES. | pLoadSeries |
| 5 | RejectLoad | .REJECT. | pLoadReject |

All files of a specified type will be loaded prior to the next file type starting to load.

Once a file is loaded, it will be renamed to indicate the load is complete.

### 4.1.3 File Rename

As mentioned, once any file is loaded, it will be renamed to indicate the load is complete. The renamed files will start with an underscore ("_") and a timestamp suffix will be appended to indicate they have been loaded. No file without an XML suffix will be loaded again.

### 4.1.4 Repetitious Loads

Administrators can govern the number of repetitions for each data file load task by specifying the number of repeats for each loader. This can be accomplished using the *Task Manager Control File*, to be described in the *System Administrator's Guide*.

### 4.1.5 Load Order Dependencies

For data with referential integrity, an order of file loading is required. Referential integrity of data files is defined in the above sections for each data file. For instance, to load the terminal data, the city data is required. If the data is not loaded in the proper order, then records will be rejected.

### 4.1.6 Load Statistics

Load statistics will be provided in the task manager log file. Statistics for both parent and child records are shown. The following statistics will be provided for each file loaded:

- Total records processed
- Total records inserted
- Total records ignored/duplicates
- Total records with errors
- Start time
- End time

### 4.1.7 Logging

One log file will be produced for data loading that will contain error messages (ERROR), warning messages (WARN), and informational messages (INFO). The system can be configured to log only certain types of messages by using the LOG_MIN_LEVEL parameter for the data services.

The parameter settings and resulting log message filtering are described below:

| LOG MIN LEVEL PARAMETER | Messages Logged |
|---|---|
| ERROR | ERROR |
| WARN | ERROR, WARN |
| INFO | ERROR, WARN, INFO |

#### *File Output Tasks*

PROS Group System will generate a confirmation/rejection remark file for each group request named "RequestID_Outstation_RequestDate".REP. This file will act as a communication media between the group user and the original requester. However, Airlines is required to provide a facility to handle the connection process.

### 4.1.8 Output Files

| Data | Pattern | Task Name |
|---|---|---|
| Request Confirmation | `RequestID_Outstation_RequestDate. REP` | |
| Request Rejection | `RequestID_Outstation_RequestDate. REP` | |
| Request Negotiation | `RequestID_HLD.xml` | |