

# INFINI Online Air Ticket発券 基本推奨ワークフロー

株式会社インフィニ トラベル インフォメーション  
2022年03月11日 更新

## セッション開始時

X-XML-Booking-Session: SESSION=START

X-XML-Booking-PCC: PCC=XXXX

X-XML-Booking-Host: HOST=SWS

## セッション終了時

X-XML-Booking-Session: SESSION=END

X-XML-Booking-PCC: PCC=XXXX

X-XML-Booking-Host: HOST=SWS



# ブランチアクセス AAA (トリプルエー) アクセス

## Session接続したPCCと異なるPCCで処理を行う場合

セッション接続後、処理を実行するPCCにAAAアクセスを行う。“ContextChangeLLSRQ”を利用



```
<ContextChangeRQ Version="X.X.X">  
  <ChangeAAA PseudoCityCode="xxx6"/>  
</ContextChangeRQ>
```

## 3 Steps



1

PNR表示

- 運賃情報 (PQ) 保存済PNRを表示

2

発券情報,支払い手段入力

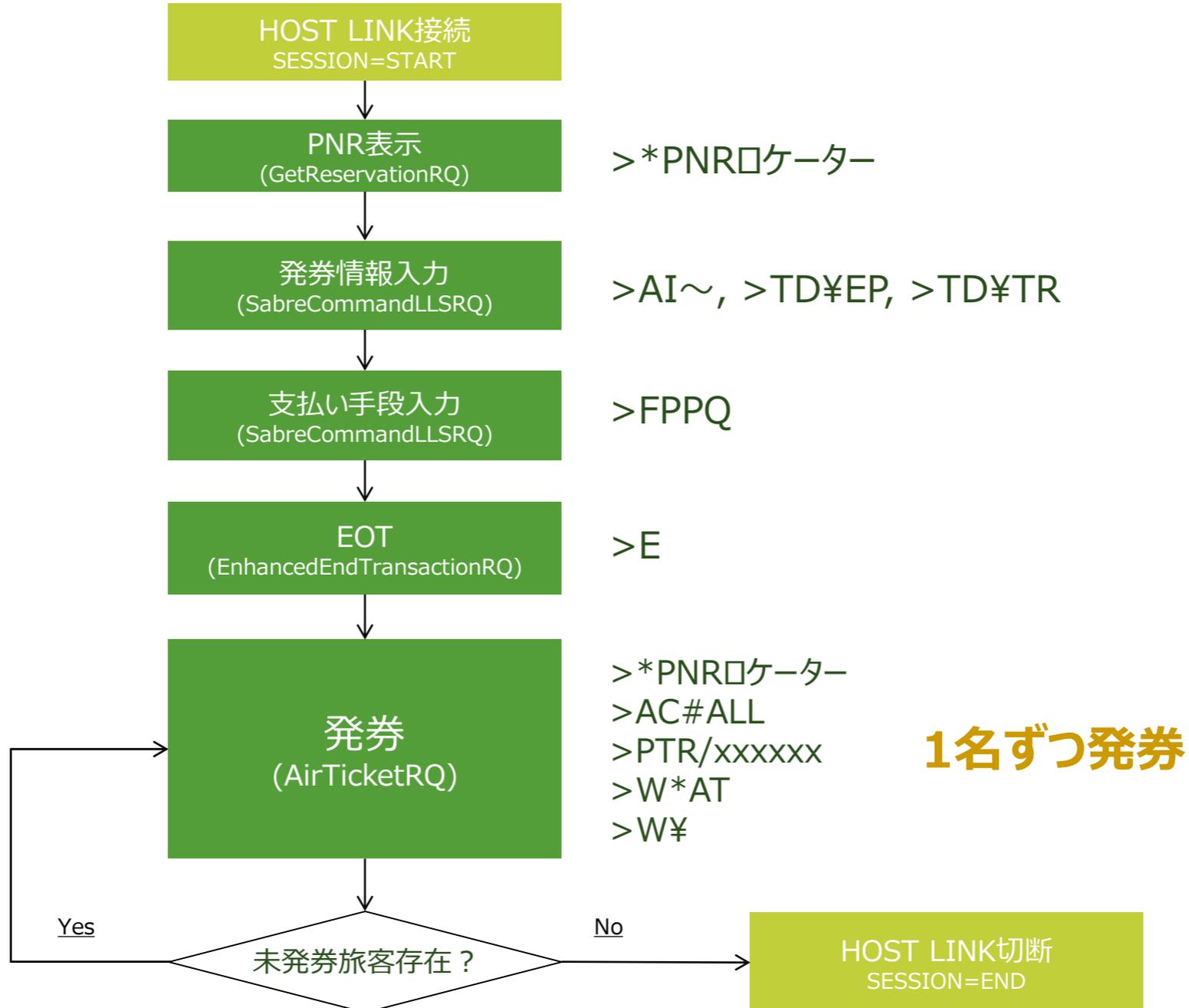
- FPライン (支払い手段、コミッション、エージェントデータ)
- アカウントコード(AI~)
- TD情報 (Endorsement、TourCode)

3

発券

- 発券プリンタ登録
- 発券実行

# 全体発券ワークフロー



# PNR 表示



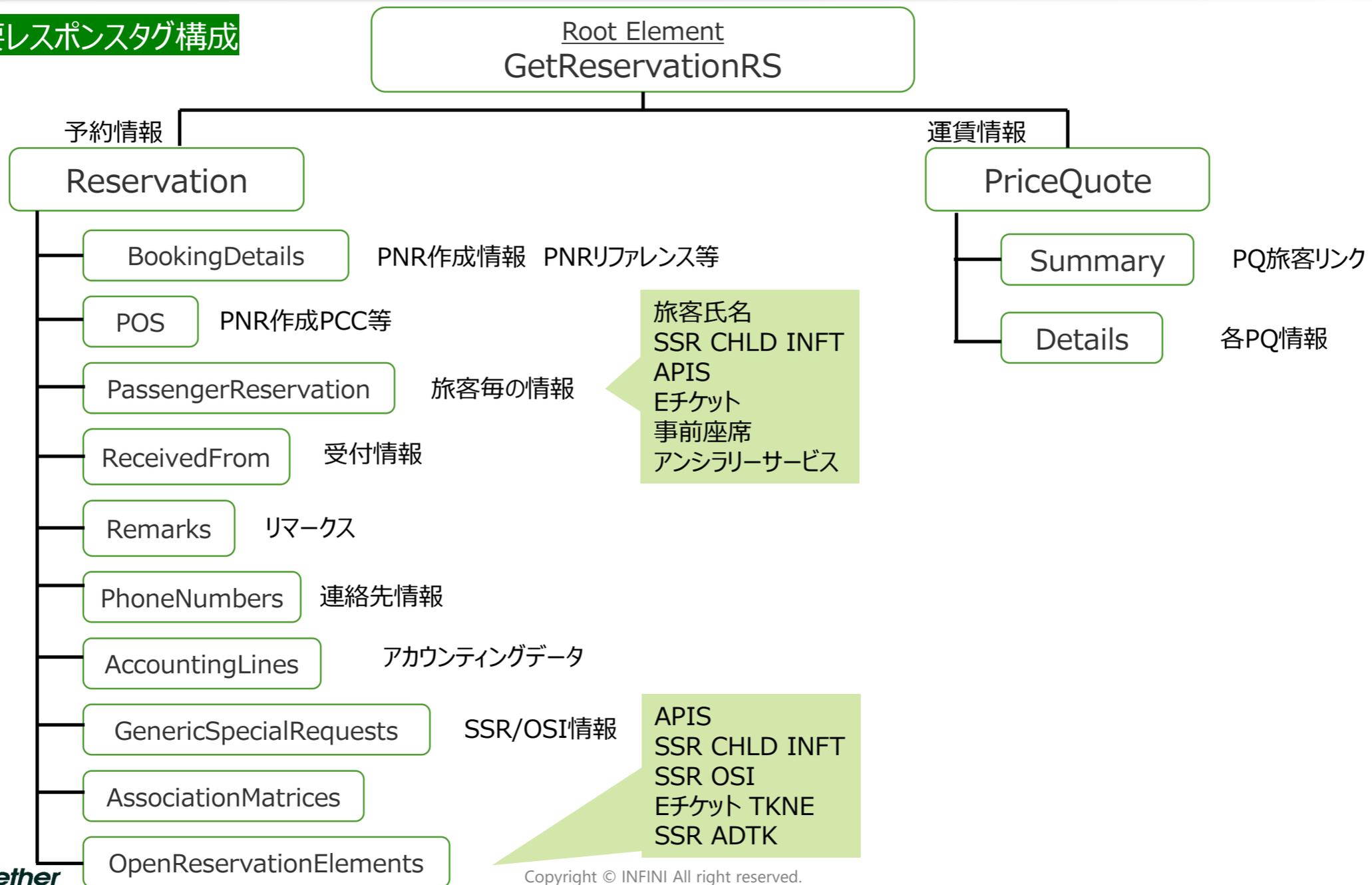
# PNR表示 GetReservationRQ リクエスト

```
<GetReservationRQ xmlns="http://webservices.sabre.com/pnrbuilder/vX_XX" Version="X.X.X">
  <Locator>QKGRHX</Locator>
  <RequestType>Stateful</RequestType>
  <ReturnOptions PriceQuoteServiceVersion="4.1.0">
    <SubjectAreas>
      <SubjectArea>PRICE_QUOTE</SubjectArea>
      <SubjectArea>INFINI_TICKET</SubjectArea>
      <SubjectArea>ADD_SSR_RPH</SubjectArea>
      <SubjectArea>ADD_SEAT_STATUS</SubjectArea>
      <SubjectArea>EXTENDED_PRS_DATA</SubjectArea>
      <SubjectArea>PASSENGER_ACCOUNT_CODE</SubjectArea>
      <SubjectArea>ADD_ACCOUNTING_LINE_NAME_NUMBER</SubjectArea>
      <SubjectArea>CARRIER_INFORMATION_CALCULATE</SubjectArea>
    </SubjectAreas>
    <ViewName>Default</ViewName>
    <ResponseFormat>STL</ResponseFormat>
  </ReturnOptions>
</GetReservationRQ>
```

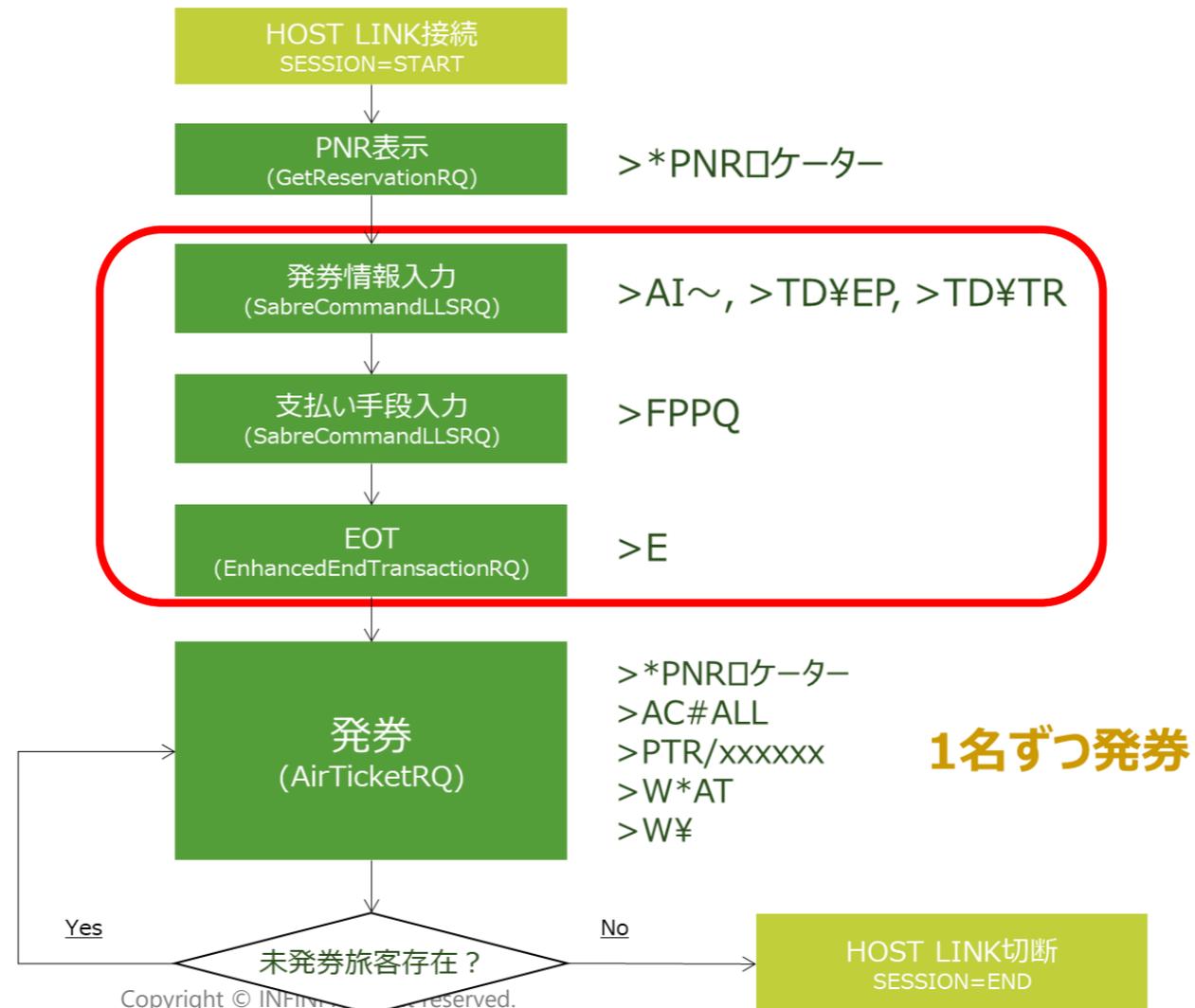
Locator : PNRリファレンス

# 1. PNR表示 GetReservationRQ レスポンス

## XML主要レスポンスタグ構成



# 発券情報・支払い手段登録



# 発券用情報入力 SabreCommandLLSRQ リクエスト

```
<SabreCommandLLSRQ Version="X.X.X" xmlns="http://webservice.sabre.com/sabreXML/2003/07">
  <Request Output="SCREEN" CDATA="true">
    <HostCommand>FPPQ1&#x00A5;FCASH&#x00A5;KP0</HostCommand>
  </Request>
</SabreCommandLLSRQ>
```

HostCommand: 発券情報入力コマンド (以下参照)

※HostCommandで使用する記号は下記の通りご指定ください

#マーク → &#x00A4;

¥マーク → &#x00A5;

入力情報	入力フォーマット
ツアーコード (TD)	<b>TD¥TR/ ツアーコード(フリーテキスト)¥Nn.n</b> 15文字までの英数字および"/","*",".","-","スペースの入力可 ¥Nに続けて旅客番号(入力がない場合、全旅客に適用)
エンドースメント (TD)	<b>TD¥EP/エンドースメント情報(フリーテキスト) ¥Nn.n</b> 60文字までの英数字および"/","*",".","-","スペースの入力可 ¥Nに続けて旅客番号(入力がない場合、全旅客に適用)
2A方式NET額	<b>W¥I2¥NJJPY50000</b> ¥Iに続けてPQレコード番号 (PQレコードが1つの場合は省略可) NJJPYに続けてNET額 (最大8桁)
アカウントコード (AI)	<b>AI¥A/2345¥C/5678¥T/123-A¥Nn.n</b> "¥"に続けてアカウントコード (R/C/S/A/D/F/G/T/K/B/Eのいずれか"¥"で複数入力可) "/"に続けてアカウントデータ (英数字、"/","-","スペースの入力可) ¥Nに続けて旅客番号(入力がない場合、全旅客に適用)
*支払い手段 (FP)	<b>FPPQ1¥FCASH¥KP0¥AD*FOP情報 (フリーテキスト)</b> PQに続けてPQレコード番号 ¥Fに続けて支払い手段 ¥KPに続けてコミッション率 ¥AD*に続けてFOP情報(任意)

\*発券必須項目

# 更新情報のPNRへの保存 EnhancedEndTransaction リクエスト

```
<EnhancedEndTransactionRQ version="X.X.X" xmlns="http://services.sabre.com/sp/eet/vX">  
  <EndTransaction Ind="true" />  
  <Source ReceivedFrom="TEST" />      @ReceiveFrom: 受付情報  
</EnhancedEndTransactionRQ>
```

# 発券



AirTicketRQで以下の処理を実行

- PNRの表示
- 発券用プリンタの登録
- アカウンティングデータの削除（再発券時に必須）
- 発券実行
- 受付情報の入力
- EOT

# 発券 AirTicketRQ リクエスト

```
<AirTicketRQ version="X.X.X" xmlns="http://services.sabre.com/sp/air/ticket/vX_X_X">
  <DesignatePrinter>
    <Printers>
      <Hardcopy LNIATA="123ABC" />
      <Ticket CountryCode="AT" LNIATA="" />
    </Printers>
  </DesignatePrinter>
  <Itinerary ID="ABCDEF" />
  <AccountingLines All="true" />
  <Ticketing>
    <MiscQualifiers>
      <FutureTicket>
        <Line NameNumber="1.1" Number="1" />
      </FutureTicket>
    </MiscQualifiers>
  </Ticketing>
  <PostProcessing>
    <EndTransaction>
      <Source ReceivedFrom="TEST" />
    </EndTransaction>
  </PostProcessing>
</AirTicketRQ>
```

Hardcopy LNIATA= : プリント番号6桁

Ticket CountryCode= : "AT"(固定)  
LNIATA= : ""(ブランク固定)

Itinerary ID= : PNRリファレンス

AccountingLines All="True" : アカウンティングラインの削除  
再発券時には必須だが、初回発券時も含めて常時"True"推奨

Line NameNumber=: 発券対象の旅客番号 **旅客1名ずつの発券を推奨**  
Line Number=: FPライン番号

Source ReceivedFrom= : 受付情報

# 発券 AirTicketRQ GhostTicketCheckの利用方法

複数旅客の発券を旅客毎に連続して順に実施する場合、稀に、内部処理エラー（航空会社からの通信遅延や処理遅延等）によって、発券が失敗する場合があります。

この事象は、AirTicketRQ Ver.1.2.1をご利用いただき、下記の推奨パラメータ（GhostTicketCheck）を設定することで、発券の失敗を軽減することが可能です。

（この設定は、APIの内部で通信や処理上の問題が発生した場合、エラーを回避する機能となります。必ずしも全てのエラーの回避が可能となるものではありませんので、予めご了承ください。）

```
<AirTicketRQ xmlns="http://services.sabre.com/sp/air/ticket/vX_X_X" version="X.X.X">
  ⋮
  (中略)
  ⋮
  <PostProcessing>
    <EndTransaction>
      <Source ReceivedFrom="INFINI"/>
    </EndTransaction>
    <GhostTicketCheck waitInterval="15000" numAttempts="6"/>
  </PostProcessing>
</AirTicketRQ>
```

# 発券 AirTicketRQ GhostTicketCheckの利用方法

PNRへ航空券番号が正常に反映された場合、AirTicketRSのSummary@committedに、trueが返却されます。

```
<AirTicketRS xmlns="http://services.sabre.com/sp/air/ticket/vX_X_X">
  :
  (中略)
  :
  <Summary committed="true">
    <DocumentNumber>1234567891234</DocumentNumber>
    <LocalIssueDateTime>2021-08-17T09:30:00</LocalIssueDateTime>
    <DocumentType>TKT</DocumentType>
    <IssuingLocation>XXX6</IssuingLocation>
    <Reservation>XXXXXX</Reservation>
    <FirstName>TAROMR</FirstName>
    <LastName>INFINI</LastName>
    <TotalAmount currencyCode="JPY" decimalPlace="0">185930</TotalAmount>
  </Summary>
</AirTicketRS>
```

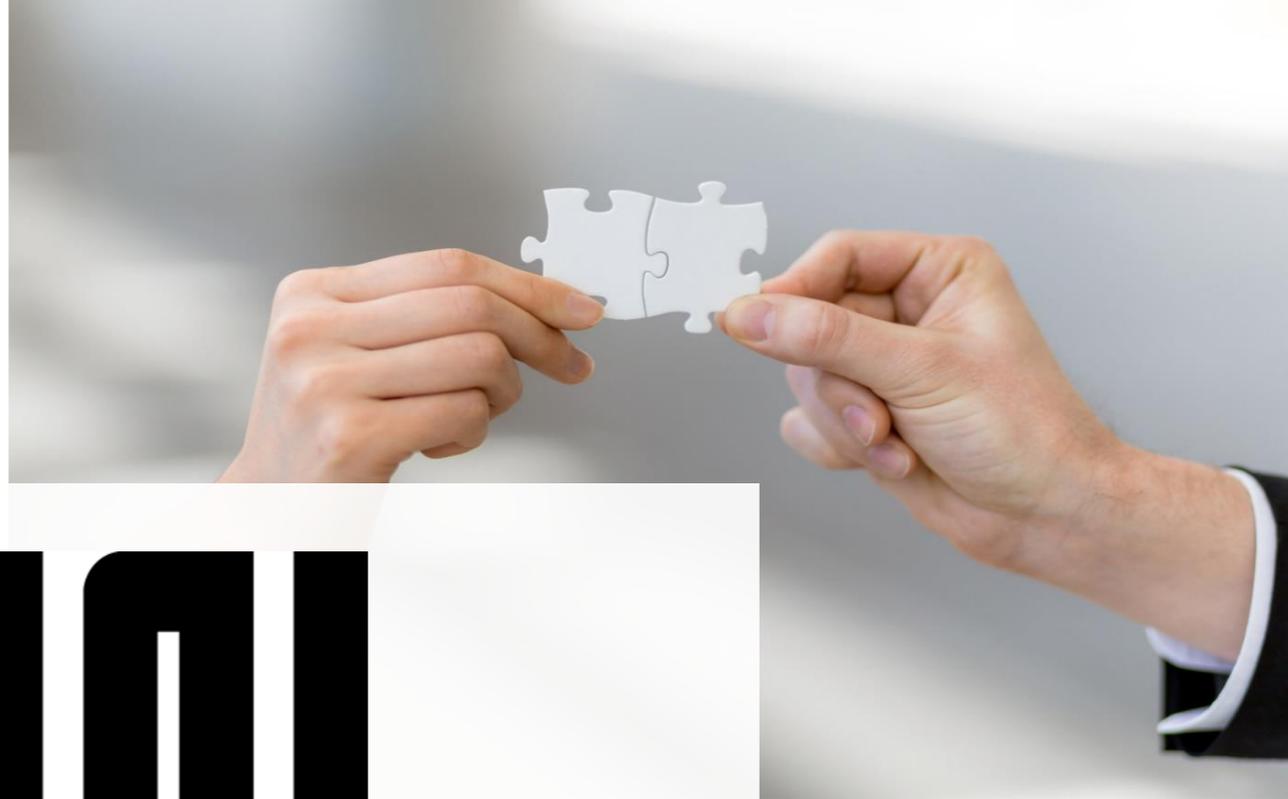
## 発券 AirTicketご利用後の確認事項

PNR表示(GetReservationRQ)とセールスレポート表示(TKT\_TravelAgencyReportsRQ)を実施し、以下3点が作成されていることをご確認ください。

いずれかの情報がない場合、発券に失敗しているため、マニュアルでのエラーハンドリングを実施する等の対応をお願いいたします。

- TEライン (GetReservationRQレスポンスの以下パスより取得)  
GetReservationRS/Reservation/PassengerReservation/TicketingInfo/TicketDetails
- アカウントングライン (GetReservationRQレスポンスの以下パスより取得)  
GetReservationRS/Reservation/AccountingLines/AccountingLine
- セールスレポート (TKT\_TravelAgencyReportsRQを利用して確認)

各APIの詳細については、INFINI Developer CenterのINFINI HOST-LINK Advance/機能仕様をご確認ください。



# INFINI

ご不明な点がございましたら[open-support@infini-trvl.co.jp](mailto:open-support@infini-trvl.co.jp)までご連絡ください。

