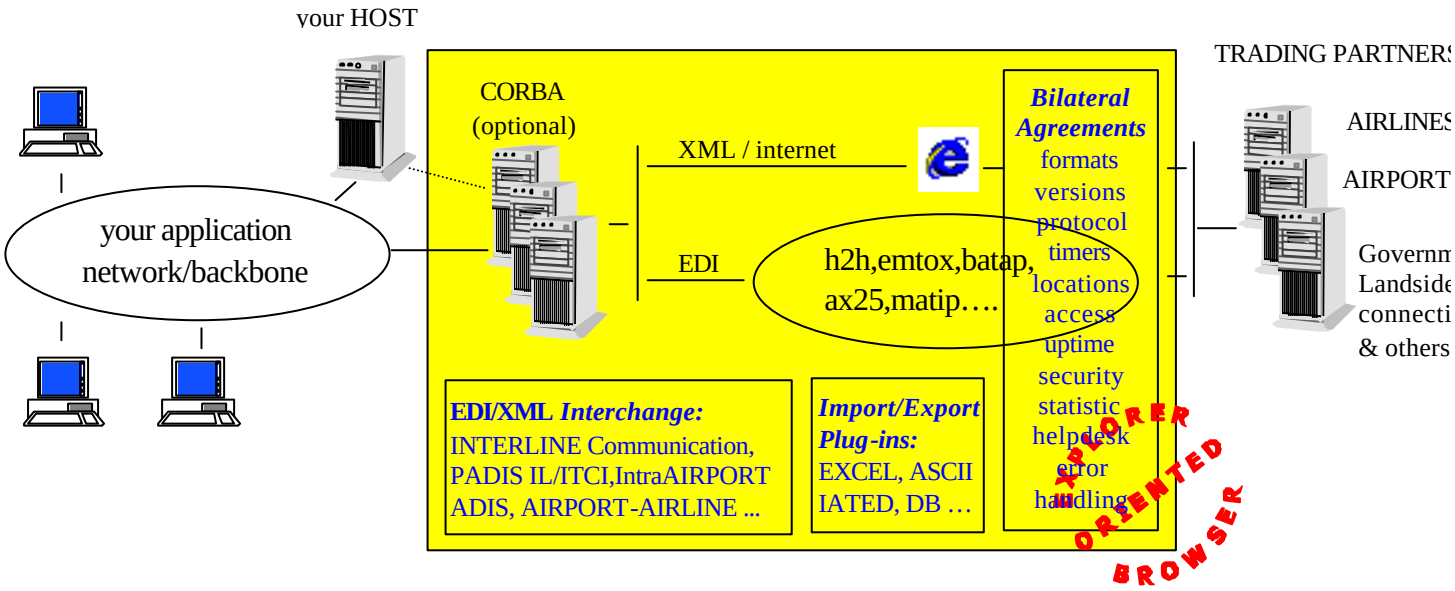
EDIX®

Generic Interactive Edifact/XML-Complex

for the Aviation Industry and *e*-Commerce business.

Win2K/NT - Server



EDIX **Generic Interactive EDIFACT/XML Complex Server**

Controls and translates single and multiple messages from an application format to a structured EDI/XML format. EDIX communicates with your trading partners over Inter-Intranet, SITA, AeroNet, or any proprietary network. EDIX is a high-performance, low-cost system.

New: US/CAN API Data lists available

EDIX has **Explorer/Browser functions** that meet your organisation's needs.

Server-Service:	translation, directory and communication service
Client-Operation:	EDIX-Process monitoring, supervision and control
Client-Admin:	Trading Partner and EDIX Management.
Intrachange-Gate:	Network/Backbone - TCP/IP, IPX/SPX, ...
Interchange-Gate:	SITA, Proprietary and Internet. Host-to-Host & legacy Protocol. TCP/IP, EMTOX, BATAP, AX25, MATIP, P1x24,.

EDIX

Server (Service):	Win2K/NT
Client:	Win2K/ME/NT/9x

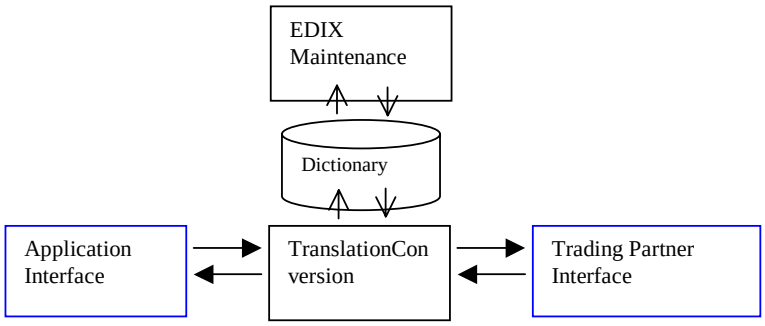
Developed with "German Perfectionism" and "Scandinavian Coolness".

EDIX interchange/message concept:

EDIX covers the several interchange and messaging concepts:

- IATCI Scenario 1: Inter Airline Through Check-in
  - Release 1: 90:1
  - Release 2: 91:1
  - Release 3: 94:1
  - Release 4: 00:1
- IATCI Scenario 2: Inter Airline Local Check-in and Flight Management
  - Release 1: 90:1
  - Release 2: 91:1
  - Release 3: 94:1
  - Release 4: 00:1
- PADIS (will be described later)
- PROPRIETARY

EDIX has following system modules and interfaces:



Application Interface:

DATAQ	Proprietary
MQ-Series	Proprietary
GDAPI	Proprietary

Trading Partner (Network) Interface:

IATCI	BATAP	X25
	MATIP	X25
	H2H	X25
	MQ-Series	
	GDAPI	

IATCI Scenario 1: Inter Airline Through Check-in:

EDIX covers the following IATCI -

INTERCHANGES (request and response):

o	DCQCKI	+	DCRCKA	Initial Through Check-in
o	DCQCKU	+	DCRCKA	Through Check-in Update
o	DCQCKX	+	DCRCKA	Through Check-in Cancel (Offload)
o	DCQPLF	+	DCRCKA	Through Check-in Passenger List Function
o	DCQBPR	+	DCRCKA	Through Check-in Boarding Pass Reprint Function
o	DCQSMF	+	DCRSMF	Seat Map Function

MESSAGES:

o	DCQCKI	DCS Through Check-in Request Query
o	DCQCKU	DCS Through Check-in Update Request Query
o	DCQCKX	DCS Through Check-in Cancel Request Query
o	DCQPLF	DCS Through Check-in Passenger List Function Query
o	DCQBPR	DCS Through Check-in Boarding Pass Reprint Function Query
o	DCQSMF	DCS Seat Map Function Query
o	DCRSMF	DCS Response for Seat Map Activity Query

Application Interface: DATAQ

DATA streaming via the use of two data queues.

Data queues can be looked at as the means to exchange data between the DCS application and other applications..
They contain data streams. These data streams are formatted according to what was sent by an application.

In this application, two data queues are defined:
DCTHRSDQ for sending requests and
DCTHRRDQ for receiving replies.

The definition of the fixed position of data fields on the data stream are as below descript.
Please note the number next to each entry is the size of the entry and then a letter specifying
M=Mandatory and C= Conditional.
When it is C and the value is not to be sent it must be sent as spaces.

DCTHRSDQ

HEADER:
The header on the data queues is always identical for all message types. This header includes:

- UNIQUE KEY 16 M

This uniquely identifies the message and is always the same through all the entries in one message.
It is constructed as below:

1. INITIATER SYSTEM 10 M
   This uniquely identifies the querying system of the message and is always the same through all the entries in one message

2. TRANSACTION NUMBER 6 M
   This uniquely identifies the message among messages from the same system.

- SAVED KEY composed of 16 C
  INITIATER SYSTEM 10 C
  TRANSACTION NUMBER 6 C
  These entries are used for the applications internal use when two DCS systems are communicating together.

- SENDER JOB (Display USER, JOB-ID) 30 M

This uniquely identifies the querying user of the message and is always the same through all the entries in one message.
This is composed of three fields of 10 characters length each.

- TARGET SYSTEM 10 C

This uniquely identifies the responding system of the message and is always the same through all the entries in one message

- TIME STAMP 14 M

This identifies the time in GMT the message was first sent.
It should follow the format: hhmmssDDMMCCYY .e.g. 06301210021971 means: Feb 10th 1971 at 06:30:12 in the morning

- MESSAGE TYPE 1 M

Message type will hold the type of the message: Currently Supported types are :
1. C=check-in request
2. P=Print
3. O=Offload request
4. S=Seat map request

- MORE FLAG 1 M

This tells whether there are more entries related to current message. (More indicator)
Values are (M=More, L=Last)

- MESSAGE SEGMENT 1 C

This tells which segment this entry belongs to :
Values are (F=Flight, P=Passenger, B=Bag tag, R=Remark)

TRAILER:
The TRAILER (rest) of the data on the data queue depends on the message currently under process.

A. Through Check-in request. Application: DCQCKI

the querying system would first send an entry containing the header with the function 'C' and indicating "more" indicator 'M' and following that the flight details as shown below:

HEADER:

……

MESSAGE TYPE	1 M	=	C
MORE FLAG	1 M	=	M

……

TRAILER:

ONWARD FLIGHT INFO (CONNECTION FLT) 23 M

THE FLIGHT THIS REQUEST IS TO BE PERFORMED ON IT IS SIMILAR TO THE FORMAT OF THE IATA DEFINITION OF THE ONWARD ENTRY.

Onward operator	3 M
Onward flight number	5 M
Onward class	1 M
Onward flight day of month	2 C
Onward flight embarkation station	3 C
Onward flight destination station	3 M
Onward flight departure time	4 C
Passenger reservation status on Onward flight	2 C

INBOUND INFO (COMING FRM FLT) 23 M

THE FLIGHT THIS PAX CONNECTING FROM. IT IS SIMILAR TO THE FORMAT OF THE IATA DEFINITION OF THE INBOUND ENTRY.

Inbound operator	3 M
Inbound flight number	5 M
Inbound class	1 M
Inbound flight day of month	2 C
Inbound flight embarkation station	3 M
Inbound flight destination station	3 C
Inbound flight departure time	4 C
Passenger reservation status on Inbound flight	2 C

COUNT OF PAX TO BE PROCESSED 3 M

After sending the flight details the querying system would start sending passengers related to the message with the more indicator "M" and the same key of the message. Please note the "more" indicator will be kept as "M" until the last passenger is sent. It will then change to 'L'. Passenger data sent is as follows:

SURNAME	45 M
MFCI (A, M, F or C)	1 M
INF FLAG (Y if pax has accompanying infant)	1 M
LOCATOR (URP)	10 M
FIRST NAME	45 M
CLASS	1 C
PAX DESG (e.g. ID00S1)	6 C
PTY/GROUP	4 C

REQUESTED SEAT OR SEAT AREA FOR ALL PAX	14 M
BAG COUNT	3 M
BAG WEIGHT	4 M
HAND BAGGAGE	3 M

After every passenger sent will optionally follow the remarks and the Bag tag information. The message segment mentioned above will be changed to R for remarks and B for baggage. Please note we expect the APIS information to received as a PSPT remark.

Passenger additional info:

SSR IATA code	4 M
Count of this SSR	2 C
OSI and details	55 C

Baggage info:

BAGGAGE TAG NUMBER	16 M
COUNT OF CONSECUTIVE TAGS	3 C
BAGGAGE LAST DESTINATION	3 C

C. Through Check-in Cancel (Offload). Application: DCQCKX

the querying system would send a message exactly similar to the check-in request. But off-course with a function of O.

HEADER:

……			
MESSAGE TYPE	1 M	=	O
……			

E. Through Check-in Boarding Pass Reprint. Application: DCQBPR

the querying system would send a message exactly similar to the check-in request. But off-course with a function of 'P'.

HEADER:

……			
MESSAGE TYPE	1 M	=	P
……			

F. Seat Map display request. Application: DCQSMF

the querying system would send a message with a function of 'S' and the MORE flag set to 'L', and following that the flight details as shown below:

HEADER:

……			
MESSAGE TYPE	1 M	=	C
MORE F	1 M	=	M
……			

TRAILER:

ONWARD FLIGHT INFO (CONNECTION FLT) 23 M

THE FLIGHT THIS REQUEST IS TO BE PERFORMED ON IT IS SIMILAR TO THE FORMAT OF THE IATA DEFINITION OF THE ONWARD ENTRY.

Onward operator	3 M
Onward flight number	5 M
Onward class	1 M
Onward flight day of month	2 C
Onward flight embarkation station	3 C
Onward flight destination station	3 M

Onward flight departure time		4 C

DCTHRRDQ

Replies on a request come on this data queue.

HEADER:

The header on the data queues is always identical for all message types and will contain the same key as was received on the send data queue.. This header includes:

- UNIQUE KEY 16 M

  This uniquely identifies the message and is always the same through all the entries in one message. It would be the same values as sent on the DCTHRSDQ.It is constructed as below:

  1. INITIATER SYSTEM 10 M
     This uniquely identifies the querying system of the message and is always the same through all the entries in one message
  2. TRANSACTION NUMBER 6 M
     This uniquely identifies the message among messages from the same system and would also hold the same value as sent on the DCTHRSDQ.

- SAVED KEY 16 C
  INITIATER SYSTEM 10 C
  TRANSACTION NUMBER 6 C
  These entries are used for the application internal use when two DCS systems are communicating together.

- SENDER JOB (JOB/USER/NBR) 30 M
  This uniquely identifies the querying user of the message and is always the same through all the entries in one message

- TARGET SYSTEM 10 C
  This uniquely identifies the responding system of the message and is always the same through all the entries in one message

- TIME STAMP 14 M
  This identifies the time in GMT the reply was sent. It should follow the format: hhmmssDDMMCCYY

- RETURN CODE AND ENTRY TYPE 4 M
  Return code is of two parts: First two letters (major code) will hold the type of the message The second two letters (minor code) tells more details and if there are more entries related to current message. (More indicator) :
  Currently Supported types are:
  For Major code
  00 Operation ended successfully.
  The minor code can be
  00 Operation ended successfully.
  01 FLIGHT DATA ENTRY
  02 B/PASS PRINT ANOTHER TO FOLLOW
  03 B/PASS PRINT LAST IN SEQUENCE
  04 MORE DATA TO COME FOR SEAT MAP DATA
  05 LAST SEAT MAP DATA ENTRY

  For Major code
  01 Operation Was not successfull
  The minor code can be
  01 PASSENGER NOT CHECKED IN
  02 FLIGHT IS CLOSED/NOT AVAILABLE
  03 PNR NOT FOUND
  04 PNR IN ERROR
  05 PNR ALREADY CHECKED-IN
  06 PNR LOCKED/NOT AVAILABLE
  07 PNR CANNOT BE SEATED / PROBLEM IN SEATING
  08 SPLITS NOT SUPPORTED
  80 INVALID REQUEST STRUCTURE
  81 REQUEST TOO OLD (MORE THAN x SECONDS)
  99 UNDEFINED REASON

  For Major code
  99 COMMUNICATION PROBLEM
  The minor code can be
  99 COMMUNICATION PROBLEM

In case a request was not successful the return code will reflect the error and the data on the data queue will hold a message explaining the error in the following manner:

FREE TEXT MESSAGE	70 M

If there were no errors, the rest of the data on the data queue depends on the message currently under process.

TRAILER:

The TRAILER (rest) of the data on the data queue depends on the message currently under process.

A. Through Check-in response. Application: DCRCKA

the responding system would first send an entry containing the return code 00 01.
To inform that this is a FLIGHT DATA ENTRY

HEADER:

……

MAJOR CODE	2 M	=	00
MINOR CODE	2 M	=	01

……

TRAILER:

The first entry would also contain the flight information to be printed on the boarding passes. This is sent in the following manner.

ONWARD INFO (CONNECTION FLT)

CARRIER NAME	30 M
OPER CODE	3 M
FLT NUMBER	5 M
FLT DAY	2 M
FLT MONTH	2 M
FLT DEST	3 M
CLASS	1 M
CLASS NUMBER (1,2,3 OR 4)	1 M
TKT CLASS	1 C
BOARDING GATE	2 C
BOARDING TIME	4 C
FLIFO (Text about flight to be printed on BP)	30 C
ALIAS	1 C

The rest of the entries from the responding system will be the passengers actually checked –in data with return codes:
00 02 To inform querying system that this is data for a passenger and ANOTHER TO FOLLOW
00 03 To inform querying system that this is the last passenger's data

The format for these entries is :

SURNAME	45 M
FIRST NAME	45 M
INF (Y/N)	1 M
LOCATOR (URP)	10 M
SEAT	3 M
SEAT ATTRIBUTE (N/S)	1 M
PAX SEQ	3 M
DUPLICATE FLAG (if this is a second B/Pass)	1 M
SSRC	4 C
MARKETING ELEMENT	23 C

C. Through Check-in Cancel (Offload). Application: DCRCKA

the responding system would send a message informing the querying system if the request completed successfully or not. This will be through the use of the return code in the DCTHRRDQ.

E. Through Check-in Boarding Pass Reprint. Application: DCRCKA

the querying system would send a message exactly similar to the check-in response.

F. Seat Map display request. Application: DCRSMF

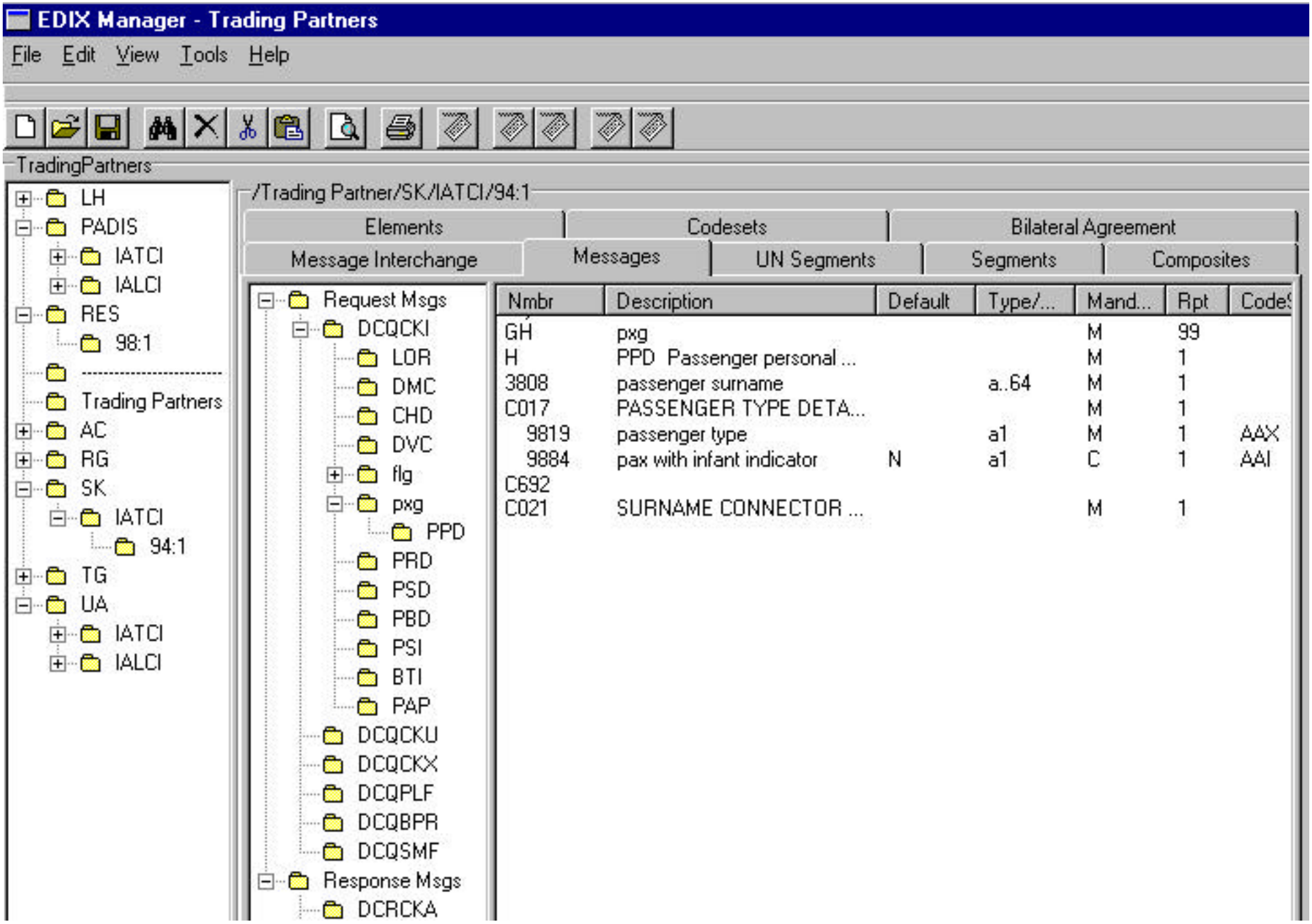
the querying system would send a message with return code

00 Request was successful.
04 More seating data to follow
05 Last entry of seating data.

The data will be sent column-by-column preceded by a letter indicating the column name.

Type of SEAT DATA		1 M
Can hold the values:		
	W= Wing column	
	1= Ruler 1	
	2= Ruler 2	
COLUMN NAMES		1 M
SEAT MAP DATA		100 M

In case of an error, code would be sent as explained earlier.


EDIX Manager - Trading Partners
File Edit View Tools Help
TradingPartners
LH
PADIS
IATCI
IALCI
RES
98:1
Trading Partners
AC
RG
SK
IATCI
94:1
TG
UA
IATCI
IALCI
/Trading Partner/SK/IATCI/94:1
Elements
Codesets
Bilateral Agreement
Message Interchange
Messages
UN Segments
Segments
Composites
Request Msgs
DCQCKI
LOR
DMC
CHD
DVC
flg
pxg
PPD
PRD
PSD
PBD
PSI
BTI
PAP
DCQCKU
DCQCKX
DCQPLF
DCQBPR
DCQSMF
Response Msgs
DCRCKA
Nmbr
Description
Default
Type/...
Mand...
Rpt
GH pxg M 99
H PPD Passenger personal ... M 1
3808 passenger surname a..64 M 1
C017 PASSENGER TYPE DETA... M 1
9819 passenger type a1 M 1 AAX
9884 pax with infant indicator N a1 C 1 AAI
C692
C021 SURNAME CONNECTOR ... M 1

XML

XML (eXtensible Markup Language) opens the door to use the generic EDIFACT standard for information exchange on the Internet.

EDIX is designed to fulfil the aviation industries need and especially the need of information exchange for Interline[1] Through Check-In and Interline in general. Nevertheless EDIX can also handle any XML information exchange business on the Internet.

For securing the XML information exchange, an encryption[2] option is provided by EDIX.

EDIX provides 2 XML interchange formats:

1) **The XML standard format**
   Where the generic EDIFACT message is completly translated/converted to an XML format, using the PADIS identifications as XML-tags.

   EDIX follows strict the "Tag" identification from the PADIS generic EDIFACT dictionary.

   In general, the XML standard format used for Interline purpose is in the existing form not so cost effective, as the implemented PADIS EDIFACT information exchange. Therefore the XML standard "Format" and "Tags" are under process to become defined by an implementation-working group under UN/OASIS. EDIX will continuously be updated due to the progress of this group.

   The XML format requires that the trading partner has implemented the PADIS identification tags and an XML parser.

2) **The XML-EDI transport format**
   Where the generic EDIFACT message is carried - as an EDIFACT message without any change - inside an XML envelope.

   The advantage - in a transition period - is that trading partners can use there existing EDIFACT parsers.

Both formats are bilateral agreeable and also the "Tags" can optionally be customised for each trading partner separately.

[1] INTERLINE means the Airline-to-Airline business, A2A -similarly to B2B (Business-to-Business).
[2] 128 bits SSL-encryption.

STANDARD FORMAT XML

EDIX provides the complete EDIFACT information exchange, according to the World-Wide-Web Consortium (W3C) common standard with use of the generic IATA PADIS common agreed format and the PADIS identifications as XML tags.

The examples bellow shows how a standard EDIFACT formatted message (example 1.0) is translated to an XML standard formatted message (example 1.1) in EDIX.

Example 1.0: Standard ***EDIFACT***[3] formatted IATCI message:

```
UNB+IATA:1+SR+DL+890701:0830+841F60
UNH+1+DCQCKI:90:1:IA+841F60
LOR+SR:GVA
FDQ+DL+573+890701+ATL+MIA++SR+120+8907011300+8907011655+ZRH+ATL
PPD+MEIER+F:Y++BARBARAMRS+MILLER:JOHN
PRD+Y
PSD+N
PBD+2:22
UNT+8+1
UNZ+1+841F60
```

Example 1.1: ***XML*** formatted IATCI message:

```
<?xml version="1.0"?>
<INTERLINE><PADIS><IATCI>
<UNB><S001><0001>IATA</0001><0002>1</0002></S001><S002><0004>SR</0004></S002>
        <S003><0010>DL</0010></S003><S004><0017>890701</0017><0018>0830</0018><S/004>
        <0020>841F60</0020></UNB>
<UNH><0062>1</0062><S009><0065>DCQCKI</0065><0052>90</0052><0054>1</0054>
        <0051>IA</0051></S009><0068>841F60</0068></UNH>
<LOR><C059><3127>SR</3127><3800>GVA</3800></C059></LOR>
<FDQ><C013><3127>DL</3127></C013><C014><3802>573</3802></C014><2281>890701</2281>
        <3215>ATL</3215><3259>MIA</3259><C015><3127>SR</3127></C015><C016>
        <3802>120</3802></C016><2281>8907011300</2281><2107>8907011655</2107>
        <3215>ZRH</3215><3259>ATL</3259></FDQ>
<PPD><3808>MEIER</3808><C017><9819>F</9819><9884>Y</9884></C017>

<3809>BARBARAMRS</3809><C018><3808>MILLER</3808><3809>JOHN</3809></C018>
        </PPD>
<PRD><C023><9800>Y</9800></C023></PRD>
<PSD><9807>N</9807></PSD>
<PBD><C027><6806>2</6806><6803>22</6803></C027></PBD>
<UNT><0074>8</0074><0062>1</0062></UNT>
<UNZ><0036>1</0036><0020>841F60</0020></UNZ>
</IATCI></PADIS></INTERLINE>
```

[3] From IATCI Documentation Version 12, Chapter: D.7.1.2 Business Case 2, Release 90:1

TRANSPORT FORMAT XML-EDI

EDIX provides an easy-to-implement Internet information exchange possibility by using XML as a transport mechanism and the existing EDIFACT definitions as the logical-protocol for information exchange.

The advantage - in a transition period and while the UN/OASIS workgroup is defining common tags and formats - is that the existing EDIFACT parsers can be used.

The examples bellow shows how a standard EDIFACT formatted message (example 2.0) is translated to an XML-EDI transport formatted message (example 2.1) in EDIX.

Example 2.0: ***Standard EDIFACT***[4] formatted IATCI message:

```
UNB+IATA:1+SR+DL+890701:0830+841F60
UNH+1+DCQCKI:90:1:IA+841F60
LOR+SR:GVA
FDQ+DL+573+890701+ATL+MIA++SR+120+8907011300+8907011655+ZRH+ATL
PPD+MEIER+F:Y++BARBARAMRS+MILLER:JOHN
PRD+Y
PSD+N
PBD+2:22
UNT+8+1
UNZ+1+841F60
```

Example 2.1: ***XML-EDI*** formatted IATCI message.

```
<?xml version="1.0"?>
<INTERLINE><PADIS><IATCI><EDIFACT>
UNB+IATA:1+SR+DL+890701:0830+841F60
UNH+1+DCQCKI:90:1:IA+841F60
LOR+SR:GVA
FDQ+DL+573+890701+ATL+MIA++SR+120+8907011300+8907011655+ZRH+ATL
PPD+MEIER+F:Y++BARBARAMRS+MILLER:JOHN
PRD+Y
PSD+N
PBD+2:22
UNT+8+1
UNZ+1+841F60
</EDIFACT></IATCI></PADIS></INTERLINE>
```

[4] From IATCI Documentation Version 12, Chapter: D.7.1.2 Business Case 2, Release 90:1