

EDIDEV KNOWLEDGE LETTER
April 30, 2018
<http://www.edidev.com>

This is part one of a multipart article about managing and processing incoming EDI files that are being sent by several trading partners. (The solution suggested here may not be suitable for all business scenarios.)

The Inbound Process - Part 1
=====

Step 1. Organize the inbound EDI files by sender - Have a folder for each trading partner, and in each of them create an INBOUND subfolder as well as a SEF subfolder. In this way, the EDI files are grouped together with the set of SEF files that will be used to process them. Example:

```
D:\App\HostCompany\  
  
  +--\TradingPartnerA\  
    +--\INBOUND\  
    +--\SEF\  
  
  +\TradingPartnerB\  
    +--\INBOUND\  
    +--\SEF\
```

Step 2. Create a Trading Partner profile table - The Trading Partner profile table will act as a look-up table for the AS2 server to direct inbound EDI files to their proper folder location. Example:

CompanyName	AS2_Id	CompanyInboundFolder
=====	=====	
TradingPartnerA	TP_AAA_1234	
D:\App\HostCompany\TradingPartnerA\EDI_INBOUND\		
TradingPartnerB	TP_BBBB_987	
D:\App\HostCompany\TradingPartnerB\EDI_INBOUND\		

KN20180401.txt

Step 3. Make the AS2 Server identify the sender of each inbound AS2 messages - The sender's AS2 identification can be obtained from the "AS2-From" header of the AS2 message. Example (in C#)

```
mailMessage oMessage = oMailDoc.GetMessageContent();
string sAS2_From = oMessage.getHeaderFieldValue("AS2-From");
```

Step 4. Once the sender's AS2 ID is obtained, look it up in the Trading Partner profile table to find its corresponding Inbound folder path. Example

```
string sSql = "select * from [TradingPartners] where AS2_CompanyId
= '" + sAS2_From.Trim() + "'";
SqlDataAdapter oDaTP = new SqlDataAdapter(sSql, oConnection);
DataSet oTPDs = new DataSet("dsTP");
oDaTP.Fill(oTPDs, "dsTP");

if (oTPDs.Tables["dsTP"].Rows.Count > 0)
{
    DataRow oTPRow = oTPDs.Tables["dsTP"].Rows[0];
    sSaveEdiFilePath = oTPRow["CompanyInboundFolder"].ToString();
}
```

Step 5. Continue 'unwrapping' the AS2 message to extract the EDI file, and then save it into the Inbound folder.

```
string sTypeMessage =
oMessage.getHeaderFieldValue("Content-Type");

if (sTypeMessage.ToLower() == "application/edi-x12") //EDI is X12
{
    oMailBody.Export(sSaveEdiFilePath + Session.SessionID.Trim() +
".X12");
}

else if (sTypeMessage.ToLower() == "application/edifact") //EDI is
EDIFACT
{
    oMailBody.Export(sSaveEdiFilePath + Session.SessionID.Trim() +
".edi");
}
```

Once the EDI file is saved in the inbound folder, the next step is to acknowledge it.

KN20180401.txt

...or maybe not?...find out in the next article. (To be continued.)

*** A C# source code for an AS2 Server that does the above can be purchased from our eShop for \$50 (AS2 client included also)

EDIDEV

"We make EDI fun!"