



IBM Software Group

# 2004 WDI / WBIC Customer Conference

*Global Business Transformation*

## Transaction Store – Why wouldn't you use it

Timothy J. Szal

*IBM Global Services*

**WebSphere.** software

*Application Innovation Services - EAI Public Sector*



 e-business software

© 2004 IBM Corporation

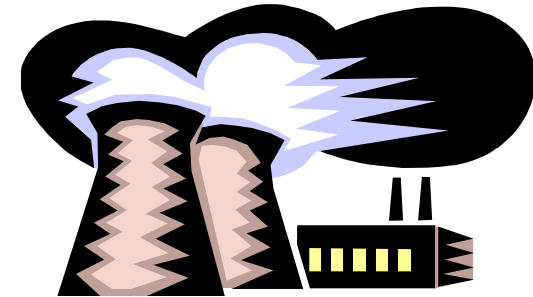
## Objectives

- A Case study of the use of WDI's Transaction Store with United Illuminating EDI system
  - How to Setup the Transaction Store
  - How to Externalize the Command file within a UNIX Script
  - How to Track Status Updates
  - Purging the Store
  - Typical Reports

## Deregulation: Breaking Out the Functional Parts

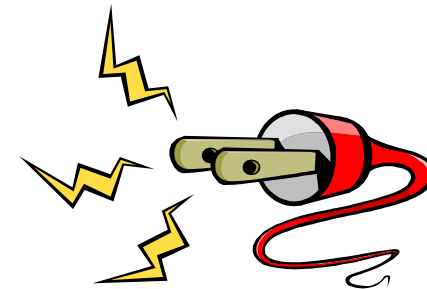
### Electric Generation Supplier (EGS)

An EGS generates power  
 They could also be brokers  
 The consumer can choose, among other things, by price, or how the power was generated



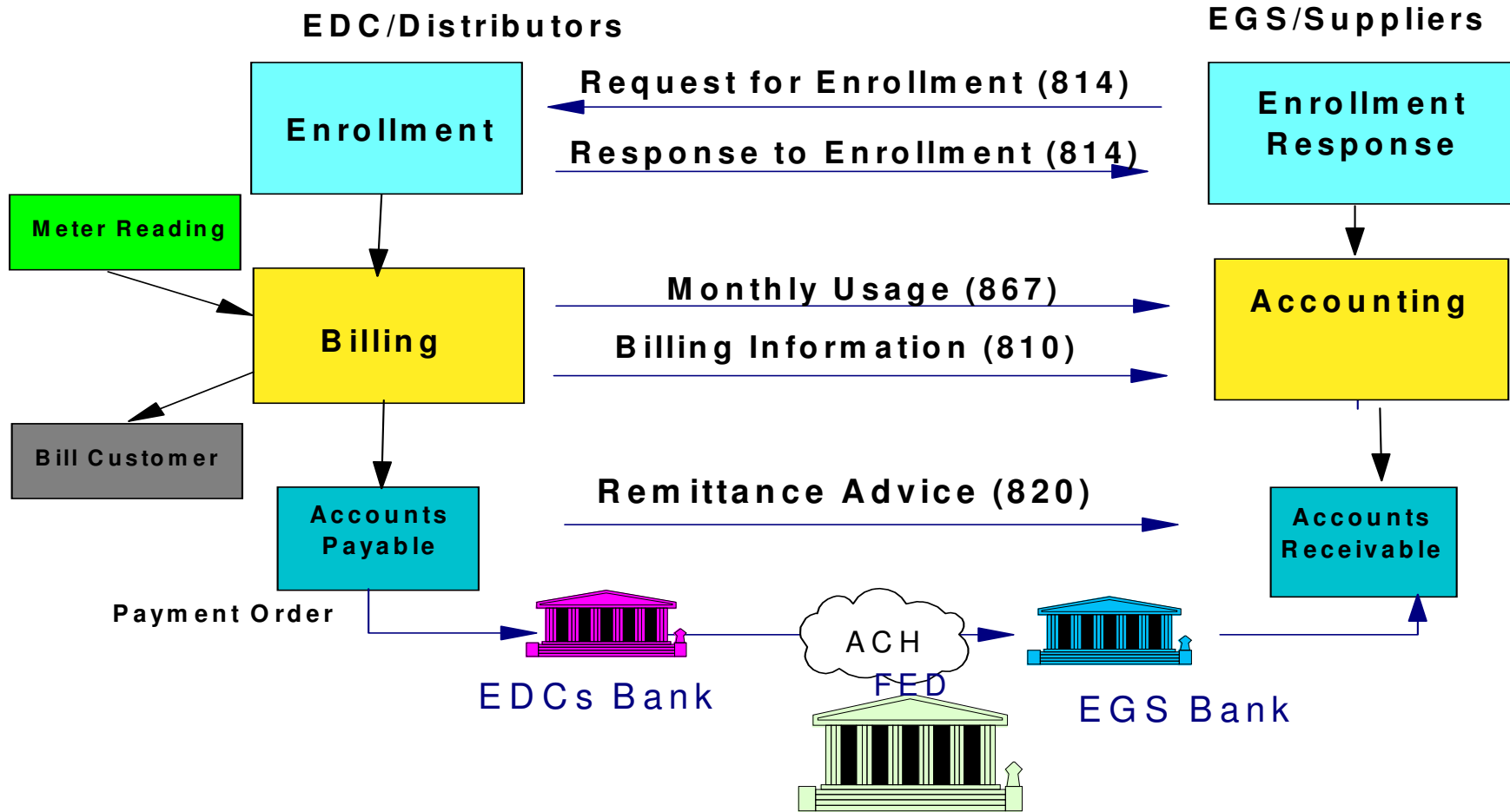
### Electric Distribution Company (EDC)

An EDC Distributes the power  
 They read the meters  
 They create the bills to the consumer



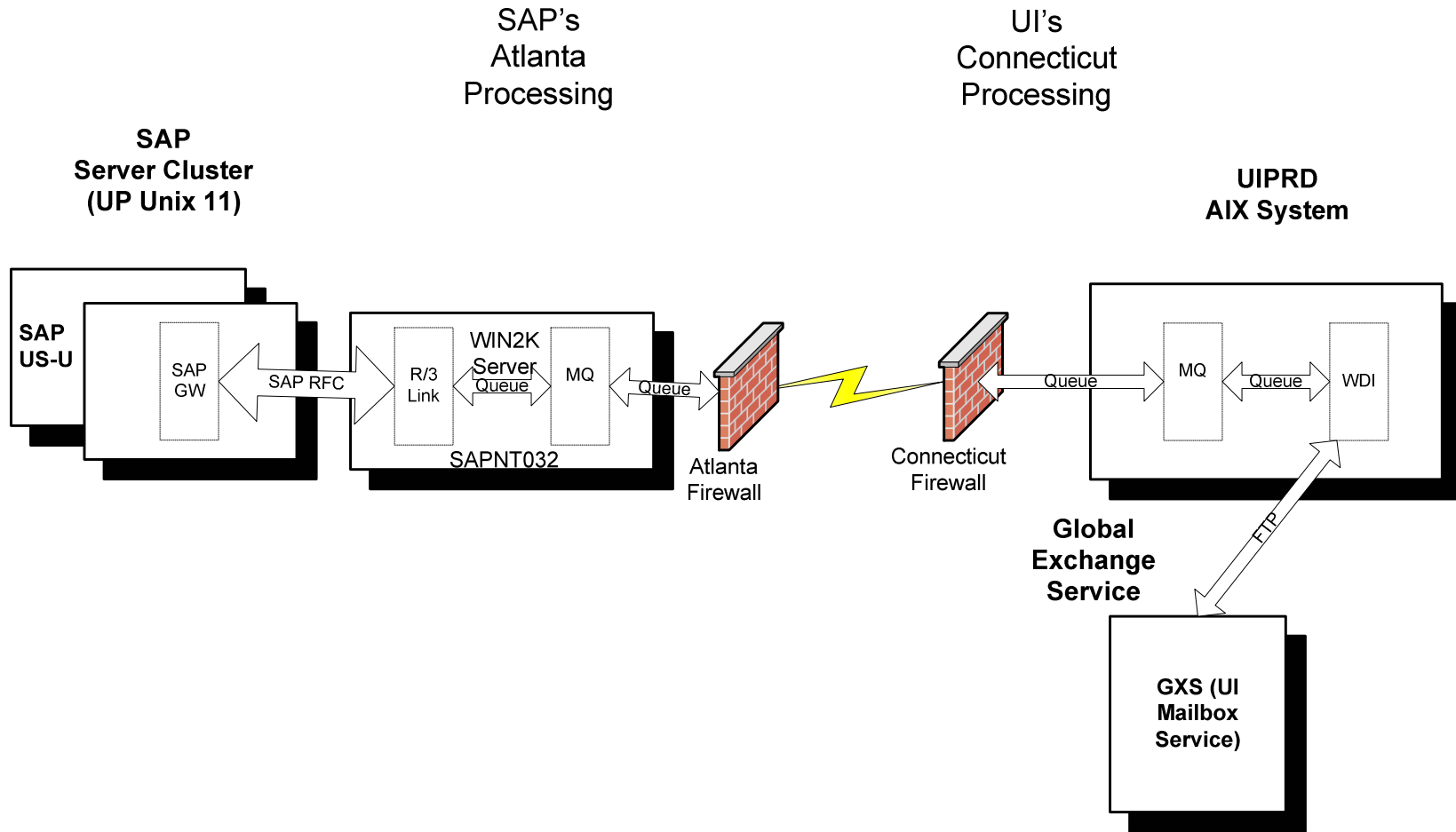
## Notes:

# Simplified Utility Industry EDI Model



## Notes:

# Production Topology for UI Case Study



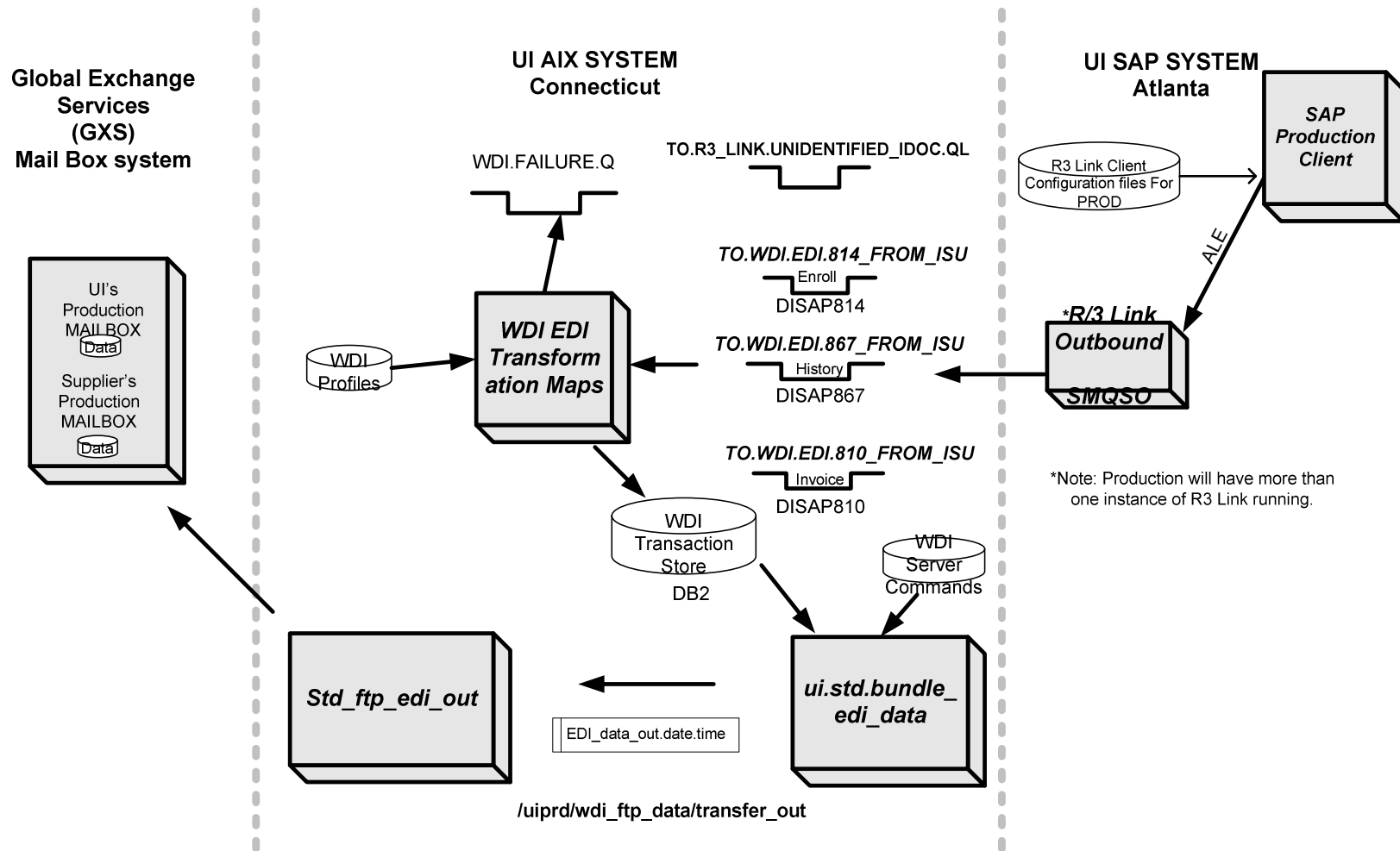
## Notes:

The architectural principals associated with the placement of application queues are:

- R/3 Link work queues will be local to the queue manager where R/3 Link resides
- R/3 Link input queues will be local to the queue manager where R/3 Link resides
- R/3 Link output queues will be local to the queue manager where WDI resides
- WDI input queues will be local to the queue manager where WDI resides, WDI output queues with messages destined for SAP will be local to the queue manager where R/3 Link resides
- Should either of the queue managers loose contact with each other (e.g. a network failure), the local application will continue processing.



# Application Queue/Script Outbound EDI Topology



## Notes:

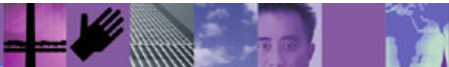
The above Figure depicts the path that all outbound EDI transactions take. EDI documents going to external trading partners will originate in SAP. As shown in above, the following process flow takes place:

ALE/SAP processes integrated with R3/Link places the EDI transaction into a queue. The placement of a document into a particular queue is based on:

- the message type
- the receiving partner ID

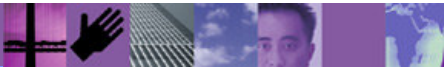
The message type is determined by the value in the message type field, and the trading partner ID is determined by the value in receiving partner field. In R3 link, the routing configuration file has to be setup correctly to route each transaction to the appropriate queue. If the configuration is not setup correctly then the transaction will go off to the unidentified IDOC queue. From this point on, the message is typed (e.g., IDOC 814 Responses, IDOC 867 Usage History)

Once a transaction arrives on a queue, MQ is configured to trigger the WDI MQ adapter. The WDI MQ adapter kicks off the service profile associated with a particular queue. The following is the service profile matrix for the various EDI documents (see next figure)



# Queue Mappings to Service Profile

Transaction	Service Profile/Queue	Service Profile command Accessed via WDI Client, Setup (Service Profiles)
814 Enrollment Response	<b>DISAP814</b>	PERFORM TRANSFORM WHERE INFILE(ADF_IN) ENVELOPE(N) DICTIONARY(IDOC_TO_EDI) DOCUMENT(GEN_814_OUT) MAPID(EDI_SAP_TO_814) SYNTAX(D) CLEARFILE(Y)
810 Invoice	<b>DISAP810</b>	PERFORM TRANSFORM WHERE INFILE(ADF_IN) ENVELOPE(N) DICTIONARY(IDOC_INVOICE) DOCUMENT(INVOICE_46C_ALE) MAPID (EDI_SAP_TO_810) SYNTAX(D) CLEARFILE(Y);
867 Historical Usage	<b>DISAP867</b>	PERFORM TRANSFORM WHERE INFILE(ADF_IN) ENVELOPE(N) DICTIONARY(IDOC_HIST_867) DOCUMENT (IDOC_867) MAPID(EDI_SAP_TO_867) SYNTAX(D) CLEARFILE(Y);



## Notes:

The service profile calls the WDI map associated with the queue and transforms the data according to the rules of the map. Note that each data transformation (DT) map is identified via the service profile. Also a trading partner has to be setup and associated with a given map (done via the WDI client).

The data transformation map, transforms the data and places the individual transaction into the transaction store. This event occurs because the parameter “ENVELOPE (N)” is set within the command. Without this command the data could get transformed and placed in a file or a queue. However, changing the “ENVELOPE” parameter “Y” will not automatically route it to a queue (other profile setups would be required).

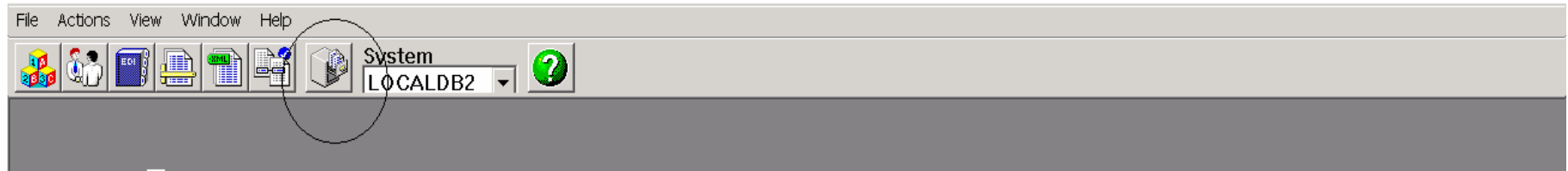
# Transaction Store Message

IDOC message placed in DISAP814 queue:	
EDI_DC40 3000EDI_DC40	6C 3012 ISU_GENERAL_REQUEST_RESPONSE
E2ISU_NAME_N1001	300000000000000012495000002000000028S UNITED ILLUMINATING
E2ISU_NAME_N1001	30000000000000001249500000300000002SJ STRATEGIC ENERGY IMITED
E2ISU_NAME_N1001	300000000000000012495000004000000028R TEST CUSTOMER
E2ISU_NAME_N1001	30000000000000001249500000500000002BT BP NAME HERE
E2ISU_ADDRESS_N3_N1000	300000000000000012495000006000000503ADDRESS FIELD 1
E2ISU_LOCATION_N4_N1000	300000000000000012495000007000000503CITY ST POSTCO
E2ISU_ITEM_IDENT_LIN000	300000000000000012495000008000000002
.....	
After translation the message is stored into the Transaction Store as:	
BGN 11 8001589197 20031219   0000000000012496	
N1 8S UNITED ILLUMINATING 1 006917967	
N1 SJ STRATEGIC ENERGY LIMITED 1 800770810	
N1 8R TEST CUSTOMER	
N1 BT  BP NAME HERE	
N3 ADDRESS FIELD 1  ADDRESS FIELD 2	
N4 CITY ST POSTCODE USA	
LIN 01 SH EL SH CE	
ASI WQ 021	
REF 7G A13	
REF 12 1000000000180	

## Notes:

**As each SAP IDOC is processed by WDI, it is placed into the Transactions Store. The Transaction Store is a collection of electronic data interchange transaction images and the control information that is needed to track the progress of transactions. Transactions enter the Transaction Store when they are translated for sending or when they are de-enveloped, after being received. All images are in standard EDI format, without envelope header and trailer segments. For example, if an Enrollment Response (814) SAP IDOC comes into the queue it is transformed and stored into the transactions store.**

# Selecting the Transaction Store via the WDI Client



**Specify Filter Values**

Query Name Find OK

Query Type Cancel

Prompt	Value
Transaction Handle Between [From Value]	
Transaction Handle Between [To Value]	
TP Nickname Like	
Translate Type (DT/SR)	
Direction (S/R)	S
EDI Standard Transaction Equal To	814
Network ID Equal To	
Internal Trading Partner ID Like	
Application Control Number Like	
Map Name Like	
Usage Indicator (T/P/I)	
Date Created Between [From Value]	
Date Created Between [To Value]	

## Transaction Store Filters

**Notes:**

**By specify selection filters translated EDI documents can be reviewed in their “pre envelope” EDI format. Most of the WDI client documentation refers to this as the “standard” format (as in ANSI standard) format.**



# Transaction Store Query Results

LOCALDB2 (Transaction Store) - Query: Find

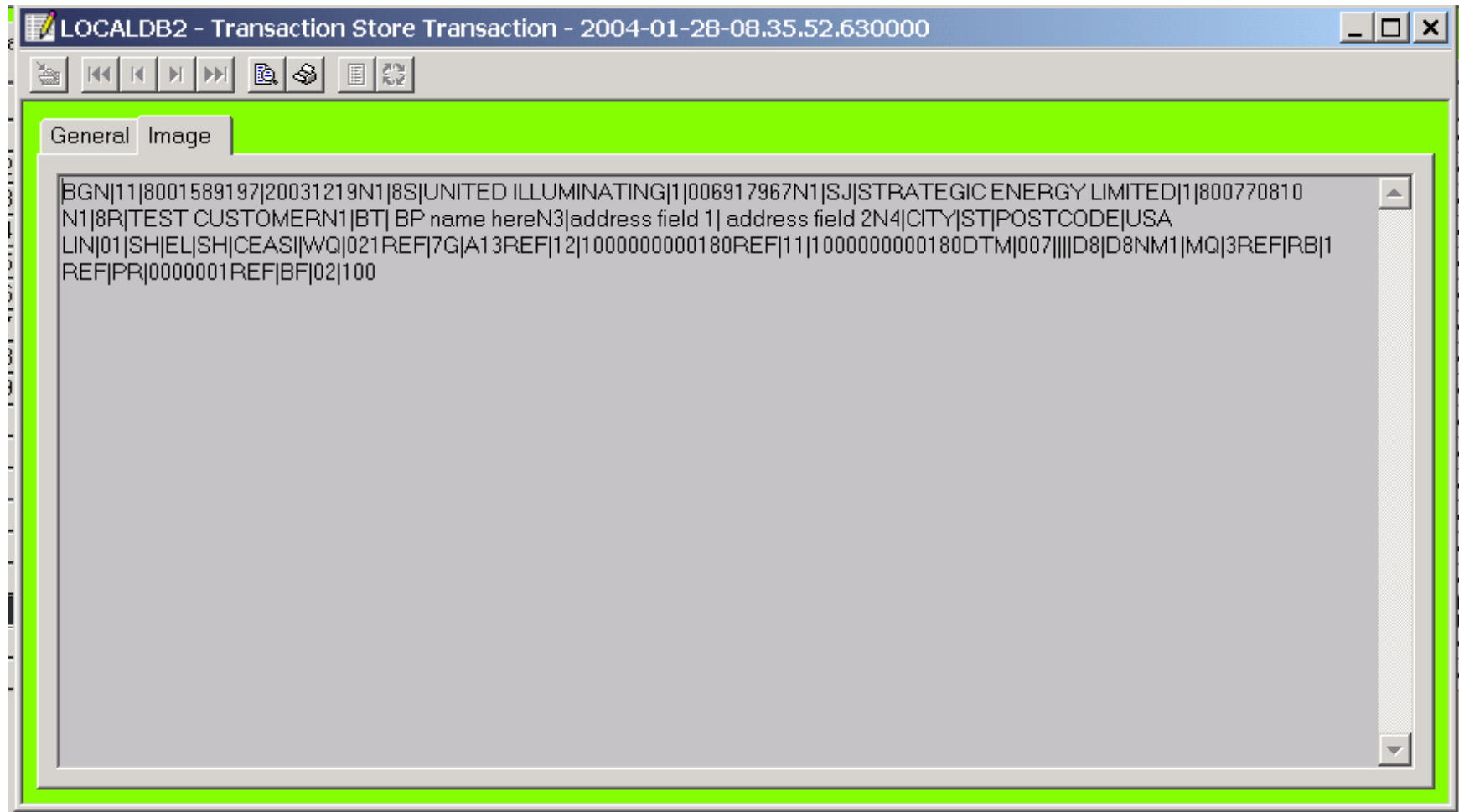
Transactions Interchanges

	Transaction Handle	Internal Trading Partner ID	Translate Type	Map Name	Date Created	Time Created
1	2004-04-07-15.20.57.040000		DT	EDI_SAP_TO_814	04/07/2004	03:20:57 PM
2	2004-04-07-15.18.14.830000		DT	EDI_SAP_TO_814	04/07/2004	03:18:14 PM
3	2004-04-07-15.16.43.950000		DT	EDI_SAP_TO_814	04/07/2004	03:16:43 PM
4	2004-04-07-15.14.31.060000		DT	EDI_SAP_TO_814	04/07/2004	03:14:31 PM
5	2004-03-26-12.57.59.840000		DT	EDI_SAP_TO_814	03/26/2004	12:57:59 PM
6	2004-02-09-17.19.16.450000		DT	EDI_SAP_TO_814	02/09/2004	05:19:16 PM
7	2004-02-09-17.18.41.920000		DT	EDI_SAP_TO_814	02/09/2004	05:18:43 PM
8	2004-02-04-21.09.13.470000		DT	EDI_SAP_TO_814	02/04/2004	09:09:13 PM
9	2004-02-04-21.05.02.620000		DT	EDI_SAP_TO_814	02/04/2004	09:05:02 PM
10	2004-02-04-20.06.25.470000		DT	EDI_SAP_TO_814	02/04/2004	08:06:25 PM
11	2004-02-04-19.45.25.470000		DT	EDI_SAP_TO_814	02/04/2004	07:45:25 PM
12	2004-02-04-19.41.15.940000		DT	EDI_SAP_TO_814	02/04/2004	07:41:16 PM
13	2004-02-01-19.58.59.860000		DT	EDI_SAP_TO_814	02/01/2004	07:58:59 PM
14	2004-02-01-19.57.25.700000		DT	EDI_SAP_TO_814	02/01/2004	07:57:26 PM
15	2004-01-28-08.40.53.610000		DT	EDI_SAP_TO_814	01/28/2004	08:40:53 AM
16	2004-01-28-08.35.52.630000		DT	EDI_SAP_TO_814	01/28/2004	08:35:53 AM
17	2004-01-26-10.13.28.400000		DT	EDI_SAP_TO_814	01/26/2004	10:13:29 AM
18	2004-01-23-08.50.26.930000		DT	EDI_SAP_TO_814	01/23/2004	08:50:28 AM

**Notes:**

**Once selections are made, the Transaction Store call will return the appropriate data**

# Transaction Store Images without EDI envelopes



**Notes:**

**Images of transactions can also be viewed before and after they have been enveloped.**

## Using Data Interchange in Command Server mode on AIX

**WDI command usage: ediservr < sample.cmd > results.txt**

**A typical command file is as follows:**

```
set plan (ediec32e);  
init;  
set file(PRTFILE,prtfile);  
set file(TRKFILE,trkfile);  
set file(EXPFILE,expfile);  
set file(XMLFILE,poxml5sr.dat);  
set file(OUTFILE,outfile);
```

```
PERFORM TRANSFORM WHERE INFILE (XMLFILE) OUTFILE(OUTFILE)  
SYNTAX(X) CLEARFILE(Y) XMLEBCDIC (N) TRACELEVEL(A2);  
term;
```

## Notes:

WebSphere Data Interchange Server always reads commands from STDIN and writes the results to STDOUT: these are treated as STREAMs. When invoked from the command line, the command line processor automatically opens STDIN and STDOUT, piping them wherever the user requests. You typically prepare a file of PERFORM commands for input and redirect the input from that file. You would probably redirect the STDOUT to a file.

An example of this usage is shown below:

```
edissrvr < sample.cmd > results.txt
```

where `sample.cmd` is the input file and has PERFORM commands, and `results.txt` contains the output from `edissrvr`. The commands files consist of a set of WebSphere Data Interchange commands separated by semicolons. Every command is terminated with a semicolon. The first command is always a SET, and the second command is always INIT. The “init” is followed by a series of Set file commands that specify the input and output files for the following PERFORM command(s).

## Contents of sample.cmd:

### **SET command**

Sets up the environment plan (ediec32e): to point to the WebSphere Data Interchange database

### **INIT command**

Loads the startup information and connects to the database using parameters defined above

### **SET FILE (LogicalFileName, RealFileName)**

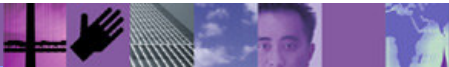
Defines various INPUT and OUTPUT files needed for translation. The LogicalFileName of each file is assigned with a real file name that includes the complete path. Depending on the type of PERFORM command used, some files are mandatory, while others are optional.

### **PERFORM**

Issues standard perform commands. This command uses LogicalFileNames for various files used.

### **TERM**

Disconnects from the database and frees all allocated memory.



## Notes:



## Perform Command Details – Transaction Store Update

**PERFORM TRANSFORM WHERE INFILE(ADF\_IN) ENVELOPE(N) DICTIONARY(IDOC\_TO\_EDI)  
DOCUMENT (GEN\_814\_OUT) MAPID(EDI\_SAP\_TO\_814) SYNTAX(D) CLEARFILE(Y)**

**ENVELOPE(Y/N)** : Default is Y. Indicates if the EDI data should be enveloped (only used for outbound EDI). If N, the transactions are just added to the store for later enveloping. **Using the “N” option which places the transaction into the store for later enveloping.**

**DICTIONARY:** This specifies the dictionary name for the input data.

**DOCUMENT:** This specifies the document name for the input data. For application data, this field is required. For EDI and XML data, this field will override values extracted from the data.

**MAPID:** This is the map name. Note the map Id contains the document name (the application layout) but, you still have to provide the document name (above).

**SYNTAX:** This specifies the syntax type for the input data. This is a required field. Valid values are:  
**D** Application data, **E** EDI data, **X** XML data

**CLEARFILE:** This indicates whether the specified file is cleared before it is used for further processing.  
Valid values are:

**Y** Clears the file before writing the translated output

**N** Does not clear the file before writing the translated output (default)

## Notes:

## UI EDI Bundle Script

### Standard Bundle EDI Script Objects

Physical Script: /usr/wdi/Dlv32/wdi\_scripts/ui.std.bundle\_edi\_data

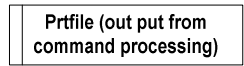
Bundle EDI Data ini File Location/Name:

/usr/wdi/Dlv32/wdi\_scripts/ini/bundle\_edi\_data.ini  
(based on ini file parameter)



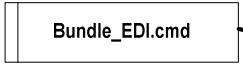
Bundle EDI Data Print File Location/Name:

/usr/wdi/Dlv32/wdi\_scripts/prtfiles/bundle\_edi\_cmds\_prtfile.txt  
(based on ini file parameter, created from prtfile)



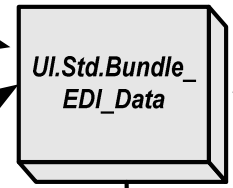
Bundle EDI Command File Location/Name:

/usr/wdi/Dlv32/wdi\_scripts/cmds/bundle\_edi.cmds  
(Command file must exist before scripts starts)



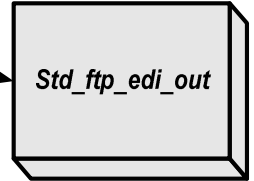
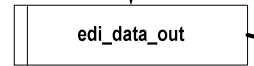
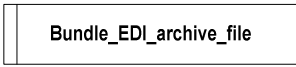
Bundle EDI Data Log File Location/Name:

/usr/wdi/Dlv32/wdi\_scripts/logs/ui.std.bundle\_edi\_data.log  
(based on ini file parameter, last command output found here)



Bundle EDI Data Archive File Location/Name:

/usr/wdi/Dlv32/wdi\_scripts/edidata\_tmp\_files  
(based on ini file parameter)



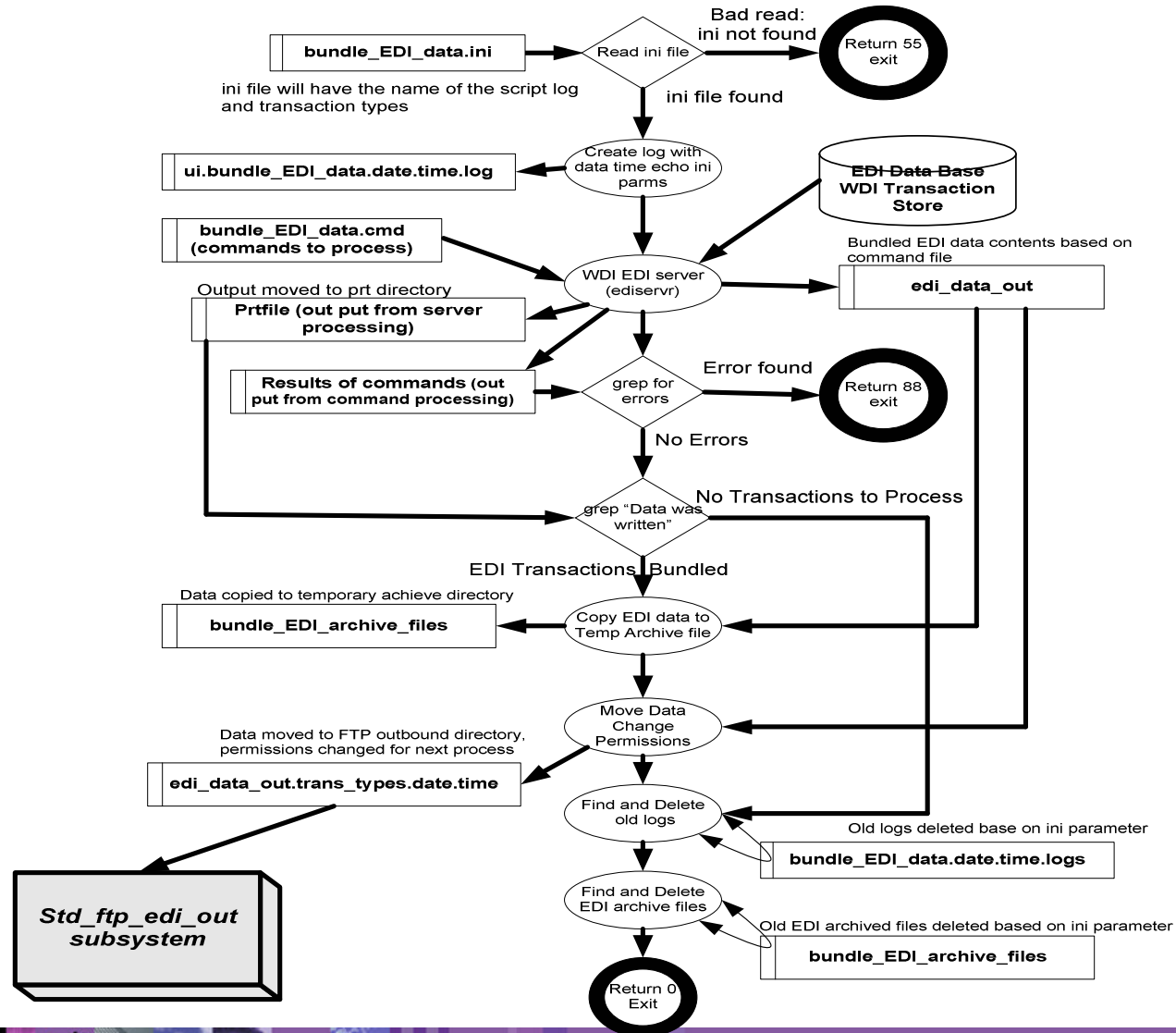
Bundled EDI Data Location/Name:

/\$uientiron/wdi\_ftp\_data/transfer\_out  
(based on ini file parameter)

**Notes:**

**Perform commands are externalized.**

### Bundle EDI Data Script (UNIX script commands - process view) Physical name: ui.std.bundle\_edi\_data



**Notes:**

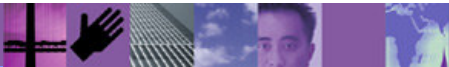
**See Unix Script Handout**

**Potential Output After (all in one file)**

PERFORM TRANSFORM  
 WHERE STDTRID(814) INTYPE(ST) DIR(S) FILEID(EDI\_OUT)  
 WHERE STDTRID(810) INTYPE(ST) DIR(S) FILEID(EDI\_OUT)  
 WHERE STDTRID(867) INTYPE(ST) DIR(S) FILEID(EDI\_OUT)

```

ISA|00|          |00|          |01|006917967      |02|178630257      |040330|1706|U|00401|000000001|0|T|^
GS|GE|178630257T|006917967T|20000110|1342|1|X|004010
ST|814|0004
BGN|11|2|20040330
N1|8S|UNITED ILLUMINATING|1|006917967
N1|SJ|SELECT ENERGY|1|178630257
N3|152 TEMPLE ST
N4|NEW HAVEN|CT|06510|US
LIN|01|SV|EL|SH|CE
ASI|WQ|021
REF|7G|A13|100
.....
NM1|MQ|3
SE|19|0004
GE|1|1
IEA|1|000000001
ISA|00|          |00|          |01|006917967T    |01|800770810STN  |040324|1247|U|00401|000000164|0|T|^
GS|IN|123456789|123456789|20040324|1247|164|X|004010
ST|810|0164
BIG|20040305|000700000129||||BD
REF|BE|1
REF|BLT|LDC
REF|BF|02
REF|12|00000001000000000504
REF|11|1000000000188
N1|8S|UNITED ILLUMINATING COMPANY|1|006917967
N1|SJ|STRATEGIC ENERGY LIMITED|1|800770810
DTM|434||||D8|20040305
IT1|1||||SV|ELECTRIC|C3|ACCOUNT||EQ|NR
.....
REF|PR|CNE 687
CTT|1
SE|23|0164
GE|1|164
IEA|1|000000164
    
```



## Notes:

In the example above you see the service segments highlighted in bold print. This is to point out that this particular perform statement will create one file that has a potential to have many different type of transactions, going to different trading partners. In the sample, there are two types of transaction; an Invoice (810) and an Enrollment (814). The service segments are created at the time the “PERFORM” commands are executed, they are seen above in bold, and again the service segments are added by WDI during the bundling process.



## ***Bundle EDI Data” and Transaction Store Status - Update***

For a typical EDI system, a transaction can the following status values:

- 20** Send translated error - status if error is encountered during the mapping process, but validation levels are set to place the data in the transaction store – should only happen during testing
- 21** Send translated - status after the data is placed in the transaction store but not bundled. This will be the typical status of pre-bundled EDI transactions
- 30** Enveloped – the status after “Bundle EDI Data” script runs (enveloped = bundled)
- 61** Transaction accepted (only when 997 from trading partner is expected)
- 62** Transaction rejected (only when 997 from trading partner is expected)
- 63** Transaction accepted with errors (when 997 from trading partner is expected)

## Notes:

# Getting to the Transaction Status

LOCALDB2 (Transaction Store) - Query: Find

Transactions Interchanges

	Transaction Handle	Internal Trading Partner ID	Translate Type	Map Name	Date Created	Time Created
1	2004-04-07-15.20.57.040000		DT	EDI_SAP_TO_814	04/07/2004	03:20:57 PM
2	2004-04-07-15.18.14.830000		DT	EDI_SAP_TO_814	04/07/2004	03:18:14 PM
3	2004-04-07-15.16.43.950000		DT	EDI_SAP_TO_814	04/07/2004	03:16:43 PM
4	2004-04-07-15.14.31.060000		DT	EDI_SAP_TO_814	04/07/2004	03:14:31 PM
5	2004-03-26-12.57.59.840000		DT	EDI_SAP_TO_814	03/26/2004	12:57:59 PM
6	2004-02-09-17.19.16.450000		DT	EDI_SAP_TO_814	02/09/2004	05:19:16 PM
7	2004-02-09-17.18.41.920000		DT	EDI_SAP_TO_814	02/09/2004	05:18:43 PM
8	2004-02-04-21.09.13.470000		DT	EDI_SAP_TO_814	02/04/2004	09:09:13 PM
9	2004-02-04-21.05.02.620000		DT	EDI_SAP_TO_814	02/04/2004	09:05:02 PM
10	2004-02-04-20.06.25.470000		DT	EDI_SAP_TO_814	02/04/2004	08:06:25 PM
11	2004-02-04-19.45.25.470000		DT	EDI_SAP_TO_814	02/04/2004	07:45:25 PM
12	2004-02-04-19.41.15.940000		DT	EDI_SAP_TO_814	02/04/2004	07:41:16 PM
13	2004-02-01-19.58.59.860000		DT	EDI_SAP_TO_814	02/01/2004	07:58:59 PM
14	2004-02-01-19.57.25.700000		DT	EDI_SAP_TO_814	02/01/2004	07:57:26 PM
15	2004-01-28-08.40.53.610000		DT	EDI_SAP_TO_814	01/28/2004	08:40:53 AM
16	2004-01-28-08.35.52.630000		DT	EDI_SAP_TO_814	01/28/2004	08:35:53 AM
17	2004-01-26-10.13.28.400000		DT	EDI_SAP_TO_814	01/26/2004	10:13:29 AM
18	2004-01-23-08.50.26.930000		DT	EDI_SAP_TO_814	01/23/2004	08:50:28 AM

## Notes:

# Displaying the Transaction Status

LOCALDB2 - Transaction Store Transaction - 2004-04-07-15.20.57.040000

General Image

Transaction Handle	2004-04-07-15.20.57.040000		
Map Name	EDI_SAP_TO_814		
Date Created	04/07/2004	Time Created	03:20:57 PM
Translate Type	DT	Direction	S
Store Status	00	Transaction Status	30
Trading Partner Profile	SEL		
Internal Trading Partner ID			
Application Control Number			
Translation Error Level			
Translation	Y		

## Notes:

The determination of the status of a transaction can be found by going to the Transaction Store as described in section “Transactions Store”. Once Transactions have been selected , “Getting to the Transaction Status”, then the transaction status can be determined by double clicking on a transaction and reviewing the Transaction Status box.

The status will change from 21 to 30 after the document has been enveloped.

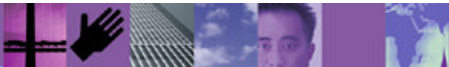
## Transaction Store a Logical and Physical Purge should be done

### Command File Logical Purge:

```
set PLAN(ediec32e);  
init;  
set file(PRTFILE,/usr/wdi/Dlv32/wdi_scripts/prtfles/edi_logical_purge_prtfile.txt);  
set file(RPTFILE,/usr/wdi/Dlv32/wdi_scripts/rpts/edi_logical_purge_rptfile.txt);  
set file(TRKFILE,/usr/wdi/Dlv32/wdi_scripts/trk/edi_logical_purge_trkfile.txt);  
PERFORM PURGE WHERE HANDLE (*-999) TO (*-30);  
term;
```

### Command File Physical Purge:

```
set PLAN(ediec32e);  
init;  
set file(PRTFILE,/usr/wdi/Dlv32/wdi_scripts/prtfles/edi_remove_transactions_prtfile.txt);  
set file(RPTFILE,/usr/wdi/Dlv32/wdi_scripts/rpts/edi_remove_transactions_rptfile.txt);  
set file(TRKFILE,/usr/wdi/Dlv32/wdi_scripts/trk/edi_remove_transactions_trkfile.txt);  
PERFORM REMOVE TRANSACTIONS WHERE HANDLE (*-999) TO (*-30);  
term;
```



## Notes:

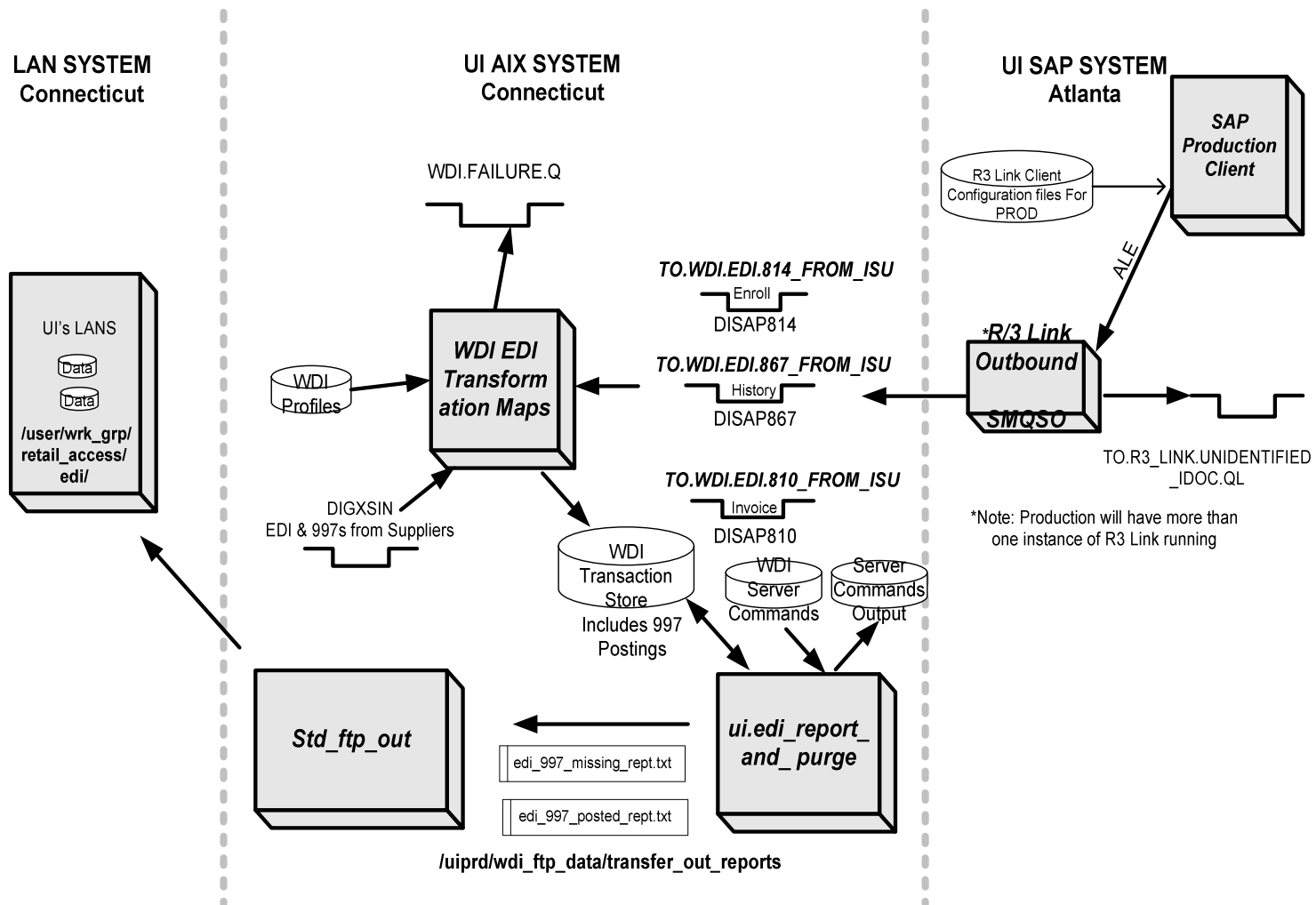
The “PERFORM PURGE WHERE HANDLE (\*-999) TO (\*-30)” selects all transactions from the Transaction Store that are thirty days old (or older). It sets a flag, which makes the transactions a logical candidate to be physically removed. The command will set this flag even if the transaction has not been acknowledged (no 997 posted against it). The “PRTFILE” will have a log of all the transactions that are selected.

Once the transaction purge flag is set, transactions will then be physically removed by the following command:

“PERFORM REMOVE TRANSACTIONS WHERE HANDLE (\*-999) TO (\*-30)”



# Application Queue/Script EDI Report Topology



## Notes:

**This shows the flow of the report generation process and the application queues that go into the topology. It also shows the report flow architecture. After the FTP send, the reports will be housed over in a LAN environment.**

**The script is used in conjunction with EDI transfer out report directory. Once the reports are placed into the WDI transfer out report directory, a standard FTP process will be used to send/FTP the reports over to the LAN.**

## Summary

- Using the Transaction Store to:
  - Collect individual messages and envelope them by:
    - Transaction Type
    - Trading partner or (Both)
  - Track status
  - Report on FA status
  - Can externalize command file for flexibility

**Notes:**

**For further information**

Timothy J. Szal    Email: [tjszal@us.ibm.com](mailto:tjszal@us.ibm.com)

Cell: (682) 553-3558