



IBM Software Group

IBM WebSphere® Data Interchange V3.3

Introduction – Data Transformation Map Trace



@business on demand.

© 2007 IBM Corporation

This presentation is an introduction to the Data Transformation Map Trace.

Agenda

- Describe the DT Trace
- Review TRACELEVEL keyword
- Trace example
- What does this mean to YOU?



The presentation will describe the Data Transformation trace, how to get the trace, and how you can use the trace.

What is the DT Trace

- Intended for use by WDI Development.
- Most WDI components that are part of the Message Flow have trace statements
- Input Buffer, Abstract Message, particular data values, document store values, etc.
- TRACELEVEL() keyword on the perform TRANSFORM command.
- Tracing begins and ends in the message flow with the WDI Message Broker.
- Some customers use the trace to trace variables within their mapping.
- Development uses the trace for problem determination.



The WDI DT Trace was initially intended for use by WDI Development. Almost all WDI components that are part of the Message Flow have trace statements that show the entry and exit of all function calls within the module that are being executed. There are also trace statements that dump out the Input Buffer, Abstract Message, particular data values, transaction store values, etc. The information recorded in the trace file depends on the TRACELEVEL() keyword on the perform TRANSFORM command.

Tracing begins and ends in the message flow with the Message Broker. The WDI Utility and WDI Data Transformation Utility do not contain tracing statements.

Some customers use the trace to trace variables within their mapping. Development uses the trace for problem determination.

TRACELEVEL Keyword

- Indicates the level of tracing done during the transform process.
- For z/OS, trace data will be written to ddname EDIDTTTRC.
- For CICS, trace data will be written to the TD queue defined for EDI standard output. If required you can change the TD queue to a TS queue.
- For AIX and Windows platforms the trace data will be written to the file defined by the environment variable EDIDTTTRC.
- You can set this using export command on AIX platforms or the set command on Windows platforms.

For example:

```
export EDIDTTTRC=trace.out
```

or

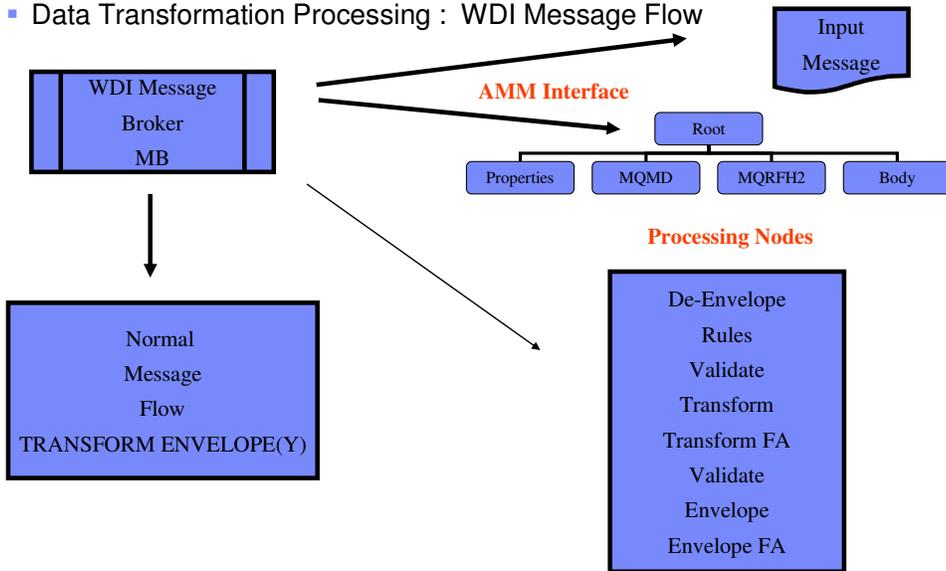
```
set EDIDTTTRC=trace.out.
```



The TRACELEVEL PERFORM keyword indicates the level of tracing done during the transform process. For z/OS, trace data will be written to ddname EDIDTTTRC. For CICS, trace data will be written to the TD queue defined for EDI standard output. If required you can change the TD queue to a TS queue. For AIX and Windows platforms the trace data will be written to the file defined by the environment variable EDIDTTTRC. You can set this using export command on AIX platforms or the set command on Windows platforms.

Data Transformation Architecture

- Data Transformation Processing : WDI Message Flow



A Normal Message Flow is the message flow for a translate and envelope process as opposed to delayed enveloping processing. It defines the processing NODES for the message. All processing nodes create and update information for the Document Store and optional record processing. Each node also has a source AMM and propagate a target AMM for the next processing node.

TRACELEVEL Keyword

- The value consists of a series of Cn values which represent the component and trace level for the component. The valid values for the component ID are:
 - ▶ A All nodes
 - ▶ D Developer node
 - ▶ E Enveloper node
 - ▶ M Message broker
 - ▶ P Parsers
 - ▶ R Rules node
 - ▶ T Transformation node
 - ▶ V Validation node



The value for the TRACELEVEL keyword consists of a series of Cn values which represent the component and trace level for the component. This is a list of the valid values for the Component.

TRACELEVEL Keyword

- The valid values for the component tracing level are:
 - ▶ 0 All trace messages are ignored.
 - ▶ 1 Normal tracing. Only the first 256 bytes of data in the buffer are written to the trace file.
 - ▶ 2 Extended tracing. The entire contents of the buffer is written to the trace file.
 - ▶ 3 Utility function tracing. Includes all the tracing done at level 2 plus additional tracing for some frequently called internal utility functions.



The value for the tracing level controls how much trace output will be produced.

TRACELEVEL Keyword

- For example,
PERFORM TRANSFORM WHERE..... TRACELEVEL(D1 V2 R2)
- Activating tracing may negatively impact performance.
- This keyword is used only with the TRANSFORM
- Customers use TRACELEVEL(T1) to trace mapping variables.



An example of the TRACELEVEL keyword value is D1 V2 R2 which would trace the Deenvelope node with normal tracing and trace the Validation and Rules nodes with extended tracing. To trace mapping variables in a Data Transformation map, you would use TRACELEVEL(T1). Tracing is normally turned off except during problem determination.

TRACELEVEL Keyword

- WDI development uses the trace for problem determination
- There is a special trace called CheckPoint Trace.
- The CheckPoint trace shows date/time and on z/OS the heap storage used. This is for tracking of memory usage.
- Not all WDI DT components have this trace. But may be added in the future.

PERFORM TRANSFORM WHERE TRACELEVEL(C1)



WDI development uses the trace for problem determination. There is a special trace called CheckPoint Trace. The CheckPoint trace shows date/time and on z/OS the heap storage used. This is for tracking of memory usage.

TRACE Example

```
----- MCB at entry: (0012c7bc), size = 580 -----  
0000:18FAE00 AD100000 48424D56 31554E49 |.1@...HBMV1UNI|  
0010:434F706 20202020 20202020 20202020 |CODE  
0020:20202020 20202020 4D5F5245 4D495854 | HBM_REMITT |  
0030:414E4345 41445844 43455F30 30312020 |ANCEADVICE_001|  
0040:20202020 58202020 20202020 20202020 | X  
0050:20202020 20200000 00006556 59656333 |...ediec3|  
0060:32653130 14D31200 41332020 20202020 |2e10.0..AS  
0070:20202020 20202020 20202020 20202020 |  
0080:20202020 20202020 20202020 20202020 |  
0090:20202020 20202020 20202020 20202020 |  
00A0:20202020 20202020 20202020 20202020 |  
00B0:20202020 20202020 20202020 20202020 |  
00C0:20202020 20202020 20202020 20202020 |  
00D0:20202020 20202020 20202020 20202020 |  
00E0:20202020 20202020 20202020 20202020 |  
00F0:20202020 20202020 20202020 20202020 |  
0100:20202020 20202020 20202020 20202020 |  
0110:20202020 20202020 20202020 20202020 |  
0120:20202020 20202020 20202020 20202020 |  
0130:20202020 20202020 20202020 20202020 |  
0140:00000000 00000000 00000000 00000000 |.....|  
0150:00000000 00000000 00000000 00000000 |.....|  
0160:00000000 00000000 00000000 00000000 |.....|  
0170:00000000 00000000 00000000 45444946 |.....EDIF|  
0180:46532020 4E4E5920 20202020 2020204E |FS NNY N|  
0190:20202020 20202020 59594E20 1E000000 | YYN ...|  
01A0:4E000000 00000000 00000000 00000000 |IN.....|  
01B0:4E000000 00000000 00000000 00000000 |IN.....|  
01C0:00000000 00000000 00000000 00000000 |.....|  
01D0:4E002020 20202020 30202059 4E4E2020 |N. O YNN|  
01E0:20202020 20202020 20202020 20202020 |  
01F0:20202020 20202020 20202020 20202020 |  
0200:20202020 20202020 20202020 20202020 |  
0210:20202020 20202020 20202020 20202020 |  
0220:20202020 20202020 20202020 2020414C | AL|  
0230:5048414F 56400000 00000000 01000000 |DUNMFM
```

Message Broker

This is a section of the Data Transformation trace. The MCB identifies the WDI Message Broker execution step.

TRACE Example

0230:5048414E 554D0000 00000000 01000000 |PHANUM.....|
 0240:00000000 |...|
 ---- State info at entry: (00000000), size = 44 ----
 EDIMB: GET THE INPUT PARMS:
 EDIMB: Dictionary name (HBMV1UNICODE)
 EDIMB: Document name (HBM_REMITTANCEADVICE_001)
 EDIMB: Syntax (X)
 EDIMB: XML Val ()
 EDIMB: Map id ()
 EDIMB: System ()
 EDIMB: Plan name (ediec32e)
 EDIMB: Trace level (A3)
 EDIMB: XML DTDS ()
 EDIMB: XML EBCDIC ()
 EDIMB: Sender Nickname ()
 EDIMB: Sender Qual ()
 EDIMB: Receiver Qual ()
 EDIMB: Receiver Nickname ()
 EDIMB: Receiver Id ()
 EDIMB: Receiver Qual ()
 EDIMB: Applid (EDIFFS)
 EDIMB: TS Active (N)
 EDIMB: TS Images (N)
 EDIMB: FA Images (Y)
 EDIMB: Earliest Env Date ()
 EDIMB: FA Delay (N)
 EDIMB: Batchset ()
 EDIMB: Proc Dup Env (Y)
 EDIMB: Envelope (Y)
 EDIMB: Deenv only (N)
 EDIMB: Direction ()
 EDIMB: Purge Int (30)
 EDIMB: MQMD/RFH2 data:
 EDIMB: Input data:
 0000:3C3F786D 6C207665 7273696F 6E3D2231 |<?xml version="1|
 0010:2E302220 656E636F 64696E67 3D225554 |.0" encoding="UT|
 0020:40308030 80808080 80808080 80808080 |...|

Message Broker

Input Message

For Help, press F1

Address Go EN 100% 11:37 AM Thursday 1/13/2005

Introduction to Data Transformation Map Trace © 2007 IBM Corporation 11

This section of the trace identifies the EDIMB: is one of the WDI Message Broker modules and also indicates WDI Message Broker execution. Input data: is the dump of the logical message that will be parsed.

TRACE Example

```

edittrc.out2 - WordPad
File Edit View Insert Format Help
EDIUCCVT: Exit cvtInit
EDIMBCNI: FactoryName (DIDEVF).
EDIMBCNI: Exit cniCreateNodeFactory
EVDCN: after cniCreateNodeFactory
EVDCN: after cvtMbs2UCS
EDIUCCVT: Enter cvtMbs2UCS
EDIMBCNI: Enter cniDefineNodeClass
EDIMBCNI: NodeName (DIDEVNode)
EDIMBCNI: Exit cniDefineNodeClass
EVDCN: after cniDefineNodeClass
EDIUCCVT: Enter cvtTerm
EDIUCCVT: Exit cvtTerm
EVDCN: Returning from bipCreateNodeFactory
EDIMB: Exit LoadFactory
EDIMB: Enter LoadFactory
EDIMB: DLL Name (edirufac)
EDIMB: Loading DLL
EDIMB: Getting dll entry point
EDIMB: Calling initialization function
EDIRU: Enter: bipGetMessageflowNodeFactory
EDIUCCVT: Enter cvtInit
EDIUCCVT: Exit cvtInit
EDIMBCNI: Enter cniCreateNodeFactory
EDIMBCNI: FactoryName (DIRULF).
EDIMBCNI: Exit cniCreateNodeFactory
EDIMBCNI: Enter cniDefineNodeClass
EDIMBCNI: NodeName (DIRULNode)
EDIMBCNI: Exit cniDefineNodeClass
EDIUCCVT: Enter cvtTerm
EDIUCCVT: Exit cvtTerm
EDIRU: Exit: bipGetMessageflowNodeFactory
EDIMB: Exit LoadFactory
EDIMB: Enter LoadFactory
EDIMB: DLL Name (edivlfac)
EDIMB: Loading DLL
EDIMB: Getting dll entry point
EDIMB: Calling initialization function
EVDCN: after bipGetMessageflowNodeFactory
  
```

Message Broker

Code Page Converter

For Help, press F1

11:41 AM
Thursday
1/13/2005

12
Introduction to Data Transformation Map Trace © 2007 IBM Corporation

EDIMBCNI: is another WDI Message Broker module and also indicates WDI Message Broker execution. EDIUCCVT: is the code page converter module.

TRACE Example

```

edittrc.out2 - WordPad
File Edit View Insert Format Help
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|
0000:55005300 52000000 65007200 74006900 |U.S.R...e.r.t.i.|
EDIMB: Enter ProcessMsgHdrs
AMMWRP: Exit cniSetInputBuffer -- Success
EDIUPAMM: Enter ammCreateContext
EDIUPAMM: Exit ammCreateContext
EDIUPAMM: Enter ammParseBuffer
EDIUPAMM: Exit ammParseBuffer
MSGUT: Enter cciGetPropertyElt
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|
0000:6D006300 64000000 2681C377 50FDAE00 |m.c.d...t.AwPy@.|
EDIMB: Enter MbDbOpen
EDIMB: Exit MbDbOpen
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|
0000:55005300 52000000 65007200 74006900 |U.S.R...e.r.t.i.|
0000:69006E00 70007500 74000000 74006900 |I.n.p.u.t...t.i.|
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|
0000:69006E00 70007500 74000000 74006900 |I.n.p.u.t...t.i.|
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|
0000:69006E00 70007500 74000000 74006900 |I.n.p.u.t...t.i.|
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|
0000:69006E00 70007500 74000000 74006900 |I.n.p.u.t...t.i.|
EDIMB: cSAfflag = N.
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|
0000:69006E00 70007500 74000000 74006900 |I.n.p.u.t...t.i.|
EDIMB: cCDFlag = N.
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|
0000:69006E00 70007500 74000000 74006900 |I.n.p.u.t...t.i.|
EDIMB: cLastMsg = Y.
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|

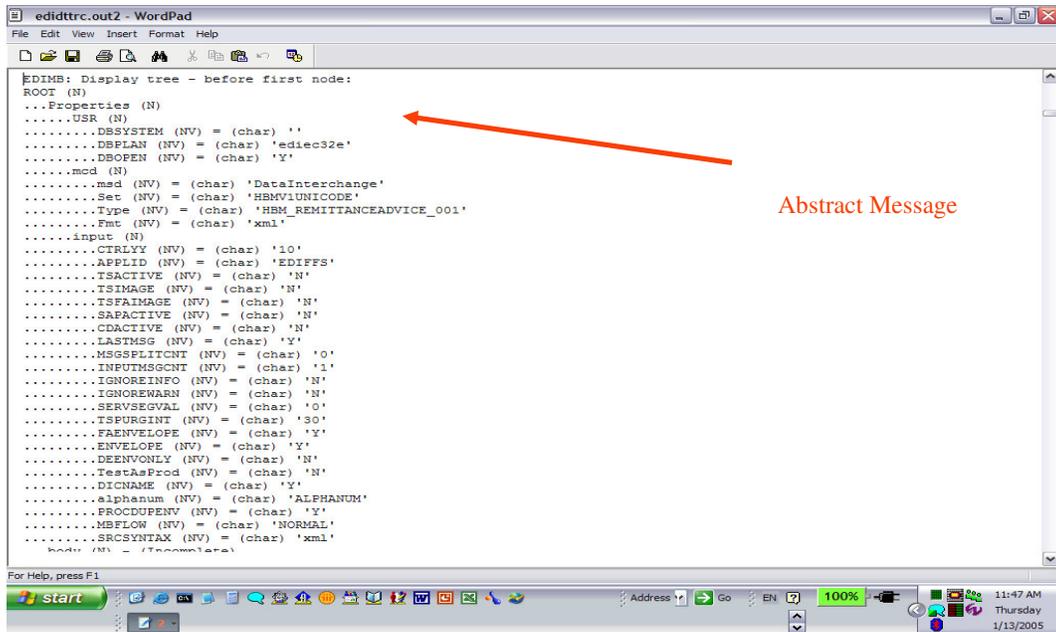
```

AMM Interface

XML Parser

AMMWRP: is the WDI Abstract Message interface. EDIUPAMM: is the WDI XML parser interface.

TRACE Example



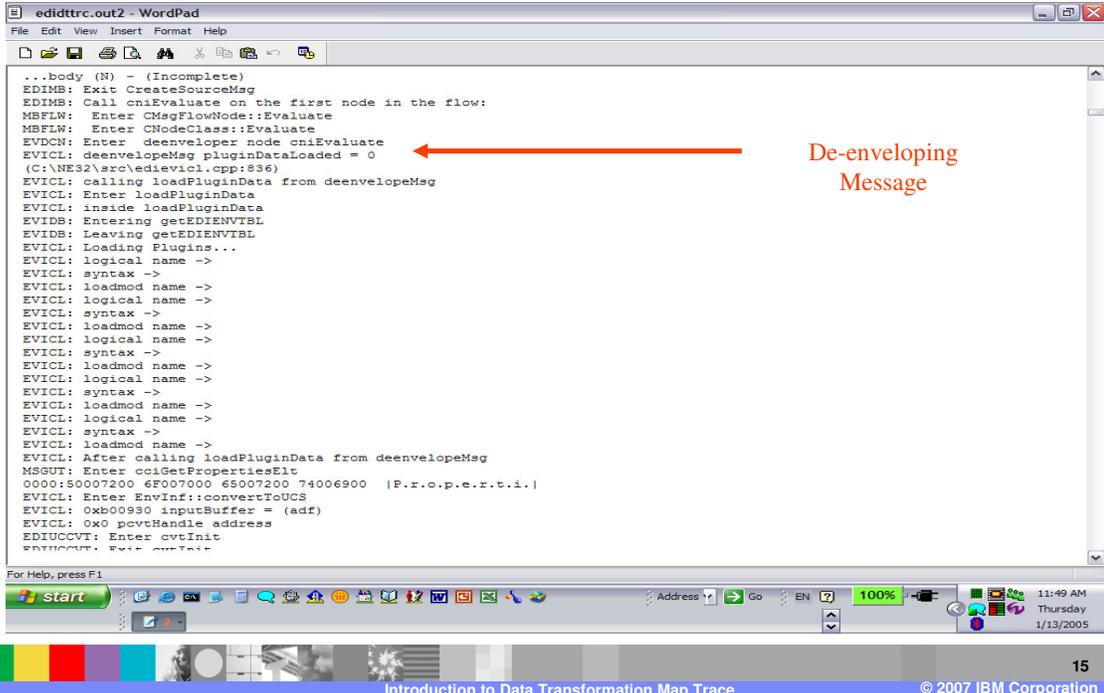
```

EDIMB: Display tree - before first node:
ROOT (N)
...Properties (N)
.....USR (N)
.....DBSYSTEM (NV) = (char) ''
.....DBPLAN (NV) = (char) 'ediec32e'
.....DBOPEN (NV) = (char) 'Y'
.....msd (N)
.....msd (NV) = (char) 'DataInterchange'
.....Set (NV) = (char) 'HBMVIUNICODE'
.....Type (NV) = (char) 'HBM_REMITTANCEADVICE_001'
.....Fmt (NV) = (char) 'xml'
.....input (N)
.....CTRLLY (NV) = (char) '10'
.....APPLID (NV) = (char) 'EDIFFS'
.....TSACTIVE (NV) = (char) 'N'
.....TSIMAGE (NV) = (char) 'N'
.....TSFIMAGE (NV) = (char) 'N'
.....SRACTIVE (NV) = (char) 'N'
.....CDACTIVE (NV) = (char) 'N'
.....LASTMSG (NV) = (char) 'Y'
.....MSGSPITCNT (NV) = (char) '0'
.....INPUTMSGCNT (NV) = (char) '1'
.....IGNOREINFO (NV) = (char) 'N'
.....IGNOREWARN (NV) = (char) 'N'
.....SERVSEGVAL (NV) = (char) '0'
.....TSPURGINT (NV) = (char) '30'
.....FAENVELOPE (NV) = (char) 'Y'
.....ENVELOPE (NV) = (char) 'Y'
.....DEENVELOPE (NV) = (char) 'N'
.....TestAsProd (NV) = (char) 'N'
.....DICNAME (NV) = (char) 'Y'
.....alphanum (NV) = (char) 'ALPHANUM'
.....PROCDUPENV (NV) = (char) 'Y'
.....MBFLOW (NV) = (char) 'NORMAL'
.....SRGSYNFAX (NV) = (char) 'xml'
body (NV) = (Tocompare)
  
```

Abstract Message

The EDIMB: Display tree is the display of the WDI Abstract Message.

TRACE Example



```
..body (N) - (Incomplete)
EDIMS: Exit CreateSourceMsg
EDIMS: Call cniEvaluate on the first node in the flow:
MBFLW: Enter CMsgFlowNode::Evaluate
MBFLW: Enter CNodeClass::Evaluate
EVDCN: Enter deenvelope node cniEvaluate
EVICL: deenvelopeMsg pluginDataLoaded = 0
(C:\MES2\src\edivcl.cpp:836)
EVICL: calling loadPluginData from deenvelopeMsg
EVICL: Enter loadPluginData
EVICL: inside loadPluginData
EVIDS: Entering getEDIENVIBL
EVIDS: Leaving getEDIENVIBL
EVICL: Loading Plugins...
EVICL: logical name ->
EVICL: syntax ->
EVICL: loadmod name ->
EVICL: logical name ->
EVICL: syntax ->
EVICL: loadmod name ->
EVICL: logical name ->
EVICL: syntax ->
EVICL: loadmod name ->
EVICL: logical name ->
EVICL: syntax ->
EVICL: loadmod name ->
EVICL: logical name ->
EVICL: syntax ->
EVICL: loadmod name ->
EVICL: After calling loadPluginData from deenvelopeMsg
MSGUT: Enter cgiGetPropertyElt
0000:50007200 6F007000 65007200 74006900 |P.r.o.p.e.r.t.i.|
EVICL: Enter EnvInf::convertToUCS
EVICL: 0xb00930 inputBuffer = (adf)
EVICL: 0x0 pcvtHandle address
EDIUCCVT: Enter cvtInit
EDIUCCVT: Enter cvtInit
```

De-enveloping Message

15
Introduction to Data Transformation Map Trace © 2007 IBM Corporation

EVICL: indicates the deenvelope node execution.

TRACE Example

```

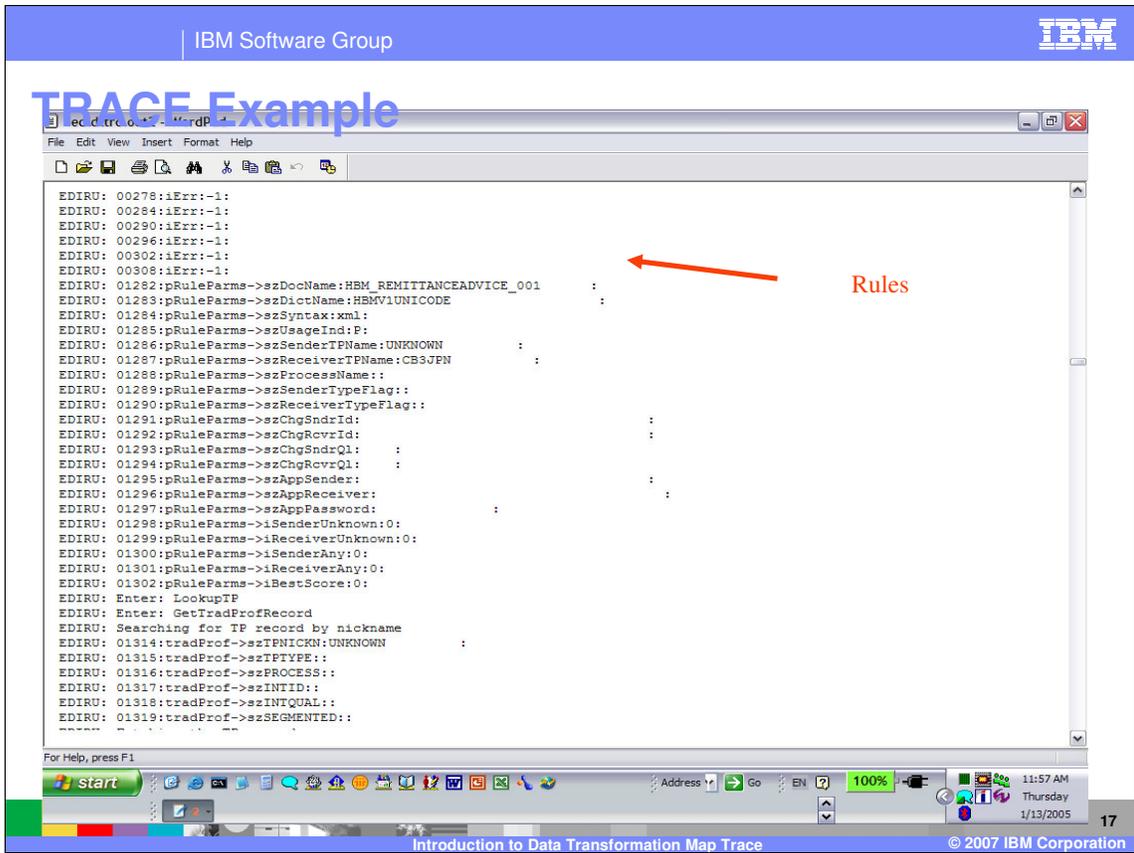
edidtrc.out2 - WordPad
File Edit View Insert Format Help
EVXML: get the Body element
EVXML: MsgRoot = af7e00
EVXML: Enter getMetaData
MSGUT: Enter cciGetPropertiesElt
0000:80007200 6F007000 65007200 74006900 |F.r.o.p.e.r.t.i.|
EVXML: Exit getMetaData
EVXML: Dictionary = HBMV1UNICODE
EVXML: Document = HBM_REMITTANCEADVICE_001
AMMNODE: Enter GetParser
AMMNODE: Checking node for parser:
...body (N) - (Incomplete)
AMMNODE: Exit GetParser
EDIUPAMM: Enter ammParseFirstChild
EDIUPAMM: Enter ParseData
EDIUPAMM: XMLBSCDIC attribute not found - using default
EDIUPAMM: Setting EBCDIC encoding on
EDIUPAMM: XMLVALIDATE attribute not found - using default
EDIUPAMM: Setting DTD on, validation off
EDIUPAMM: XMLNS attribute not found - using default
EDIUPAMM: Setting namespaces off
EDIUPAMM: XMLSCHEMAVAL attribute not found - using default no
EDIUPAMM: Setting schema validation off
EDIUPXML: Enter ParseBuffer
0000:3C3F786D 6C207665 7273696F 6E3D2231 |<?xml version="1|
0010:2E302220 656E636F 64696E67 3D225554 |.0" encoding="UT|
0020:462D3822 3F3E0D0A 3C48424D 5F52454D |F-S"?>.<HBM_REM|
0030:49545441 4E434541 44564943 455F3030 |ITTANCEADVICE_00|
0040:313E0D0A 20203C47 656E6572 616C4D65 |1>.. <GeneralMe|
0050:73736167 65486561 6465723E 0D0A2020 |essageHeader...|
0060:20203C52 65636569 76657254 5049443E |<ReceiverID>|
0070:4342393C 2F526563 65697665 72545049 |CB3</ReceiverID|
0080:443E0D0A 20202020 3C456E76 656C6F70 |ID>.. <Envelop|
0090:653E0D0A 20202020 20203C54 7970653E |e>.. <Type>|
00A0:4144463C 2F547970 653E0D0A 20202020 |ADF</Type>..|
00B0:20203C52 65636569 76657249 443E0D0A |<ReceiverID>..|
00C0:20202020 20202020 3C547970 653E4E2F |<Type>N/|
00D0:413E0D0A 20203C47 656E6572 616C4D65 |1>.. <GeneralMe|

```

XML De-enveloper

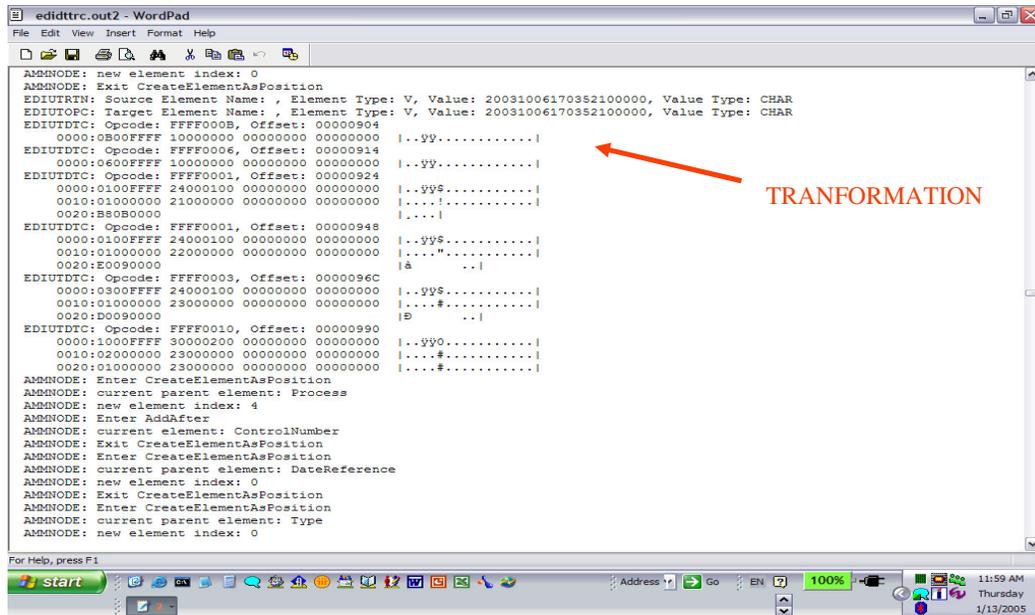


EVXML: identifies the deenveloper assigned is XML.



EDIRU: identifies the Rules node execution.

TRACE Example



```

editttrc.out2 - WordPad
File Edit View Insert Format Help
AMMNODE: new element index: 0
AMMNODE: Exit CreateElementAsPosition
EDIUTRTN: Source Element Name: , Element Type: V, Value: 20031006170352100000, Value Type: CHAR
EDIUTOPC: Target Element Name: , Element Type: V, Value: 20031006170352100000, Value Type: CHAR
EDIUTDTC: Opcode: FFFF000B, Offset: 00000904
0000:0B00FFFF 10000000 00000000 00000000 |..YY.....|
EDIUTDTC: Opcode: FFFF0006, Offset: 00000914
0000:0600FFFF 10000000 00000000 00000000 |..YY.....|
EDIUTDTC: Opcode: FFFF0001, Offset: 00000924
0000:0100FFFF 24000100 00000000 00000000 |..YYS.....|
0010:01000000 21000000 00000000 00000000 |.....|
0020:B80B0000 |...|
EDIUTDTC: Opcode: FFFF0001, Offset: 00000948
0000:0100FFFF 24000100 00000000 00000000 |..YYS.....|
0010:01000000 22000000 00000000 00000000 |....*.....|
0020:E0090000 |à ..|
EDIUTDTC: Opcode: FFFF0003, Offset: 0000096C
0000:0300FFFF 24000100 00000000 00000000 |..YYS.....|
0010:01000000 23000000 00000000 00000000 |.....|
0020:D0090000 |D ..|
EDIUTDTC: Opcode: FFFF0010, Offset: 00000990
0000:1000FFFF 30000200 00000000 00000000 |..YYO.....|
0010:02000000 23000000 00000000 00000000 |.....#.....|
0020:01000000 23000000 00000000 00000000 |.....#.....|
AMMNODE: Enter CreateElementAsPosition
AMMNODE: current parent element: Process
AMMNODE: new element index: 4
AMMNODE: Enter AddAfter
AMMNODE: current element: ControlNumber
AMMNODE: Exit CreateElementAsPosition
AMMNODE: Enter CreateElementAsPosition
AMMNODE: current parent element: DateReference
AMMNODE: new element index: 0
AMMNODE: Exit CreateElementAsPosition
AMMNODE: Enter CreateElementAsPosition
AMMNODE: current parent element: Type
AMMNODE: new element index: 0

```

EDIUTRTN;, EDIUTOPC;, and EDIUTDTC: identify the Transform node execution.

TRACE Example

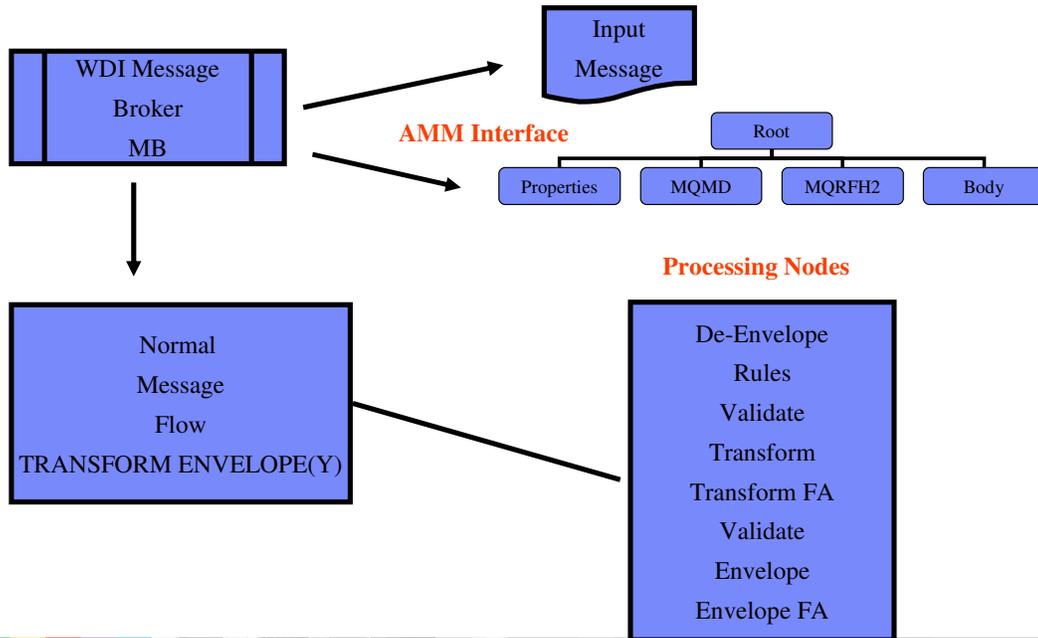
```

SDSF OUTPUT DISPLAY KANDELBP JOB04112 DSID 110 LINE 0 COLUMNS 02- 81
COMMAND INPUT ==> _
***** TOP OF DATA *****
CheckPoint 2005/01/12 21:50:49 Heap Used:36211800 ISA Count: 1 File:EDIEVX12 Lin
CheckPoint 2005/01/12 21:50:51 Heap Used:36216272 Grp Count: 1 File:EDIEVX12 Lin
CheckPoint 2005/01/12 21:50:51 Heap Used:36219008 Trx Count: 1 File:EDIEVX12 Lin
CheckPoint 2005/01/12 21:50:52 Heap Used:36499480 DTC Count: 1 File:EDIUTDTC Lin
CheckPoint 2005/01/12 21:50:53 Heap Used:36637920 DTC Count: 1 File:EDIUTDTC Lin
CheckPoint 2005/01/12 21:51:29 Heap Used:39007152 Trx Count: 101 File:EDIEVX12 L
CheckPoint 2005/01/12 21:51:31 Heap Used:39067896 DTC Count: 101 File:EDIUTDTC L
CheckPoint 2005/01/12 21:52:14 Heap Used:40629688 Trx Count: 201 File:EDIEVX12 L
CheckPoint 2005/01/12 21:52:16 Heap Used:40684744 DTC Count: 201 File:EDIUTDTC L
CheckPoint 2005/01/12 21:52:47 Heap Used:41759608 Trx Count: 301 File:EDIEVX12 L
CheckPoint 2005/01/12 21:52:48 Heap Used:41813360 DTC Count: 301 File:EDIUTDTC L
CheckPoint 2005/01/12 21:53:19 Heap Used:42895104 Trx Count: 401 File:EDIEVX12 L
CheckPoint 2005/01/12 21:53:21 Heap Used:42966720 DTC Count: 401 File:EDIUTDTC L
CheckPoint 2005/01/12 21:53:51 Heap Used:44488216 Trx Count: 501 File:EDIEVX12 L
CheckPoint 2005/01/12 21:53:58 Heap Used:44618904 DTC Count: 501 File:EDIUTDTC L
CheckPoint 2005/01/12 21:54:43 Heap Used:45768864 Trx Count: 601 File:EDIEVX12 L
CheckPoint 2005/01/12 21:54:46 Heap Used:45854512 DTC Count: 601 File:EDIUTDTC L
F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=RETRIEVE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
MA b 04/021

```

This is an example of the CheckPoint trace.

What does this mean to YOU?



This is the WDI Data Transformation Message Flow. With knowledge of the Message Flow, WDI Components, and Modules, you can identify user errors, data problems, PMRs and APARs, and maybe find a work around more accurately.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
IBM (logo)
e/Logo/business
AIX

CICS
Cloudscape
DB2
DB2 Universal Database

IMS
Informix
iSeries
Lotus

WMO
OS/390
OS/400
pSeries

Tivoli
WebSphere
xSeries
zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.