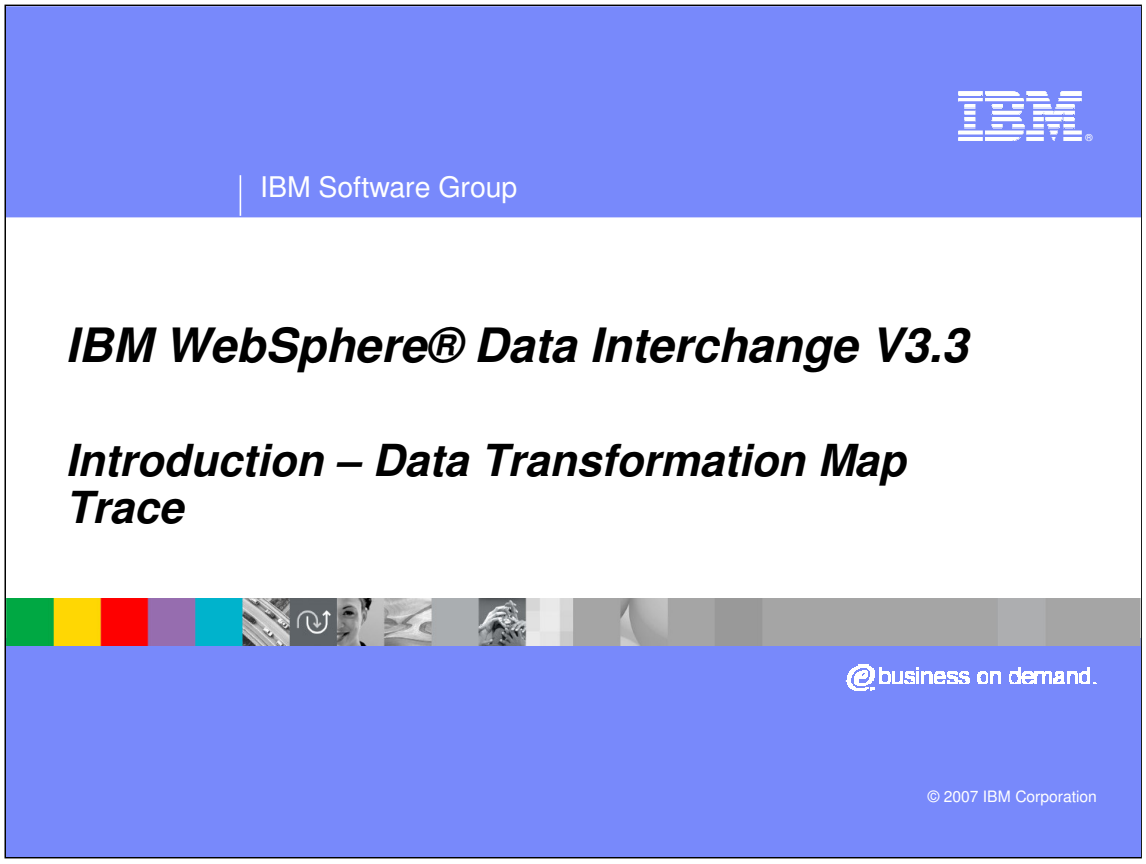


This presentation is an introduction to the Data Transformation Map Trace.

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.

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 EDIDTTRC.
- 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 EDIDTTRC.
- You can set this using export command on AIX platforms or the set command on Windows platforms.

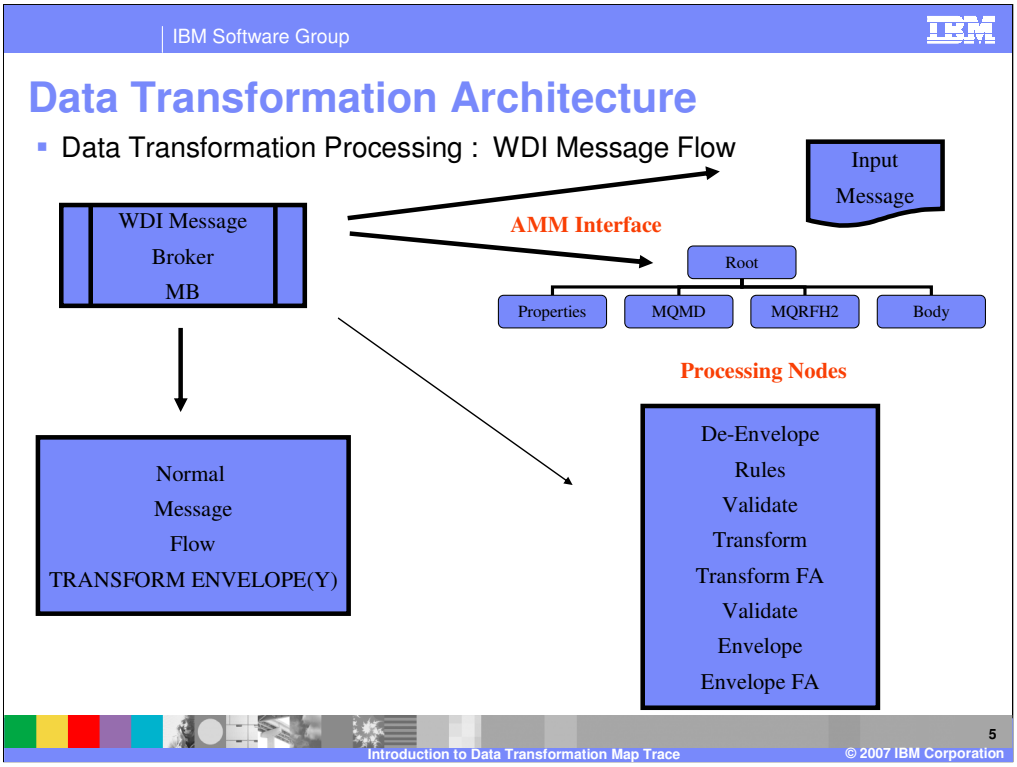
For example:

export EDIDTTRC=trace.out

or

set EDIDTTRC=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 EDIDTTRC. 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 EDIDTTRC. You can set this using export command on AIX platforms or the set command on Windows platforms.



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 Deenveloper 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 byes 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.

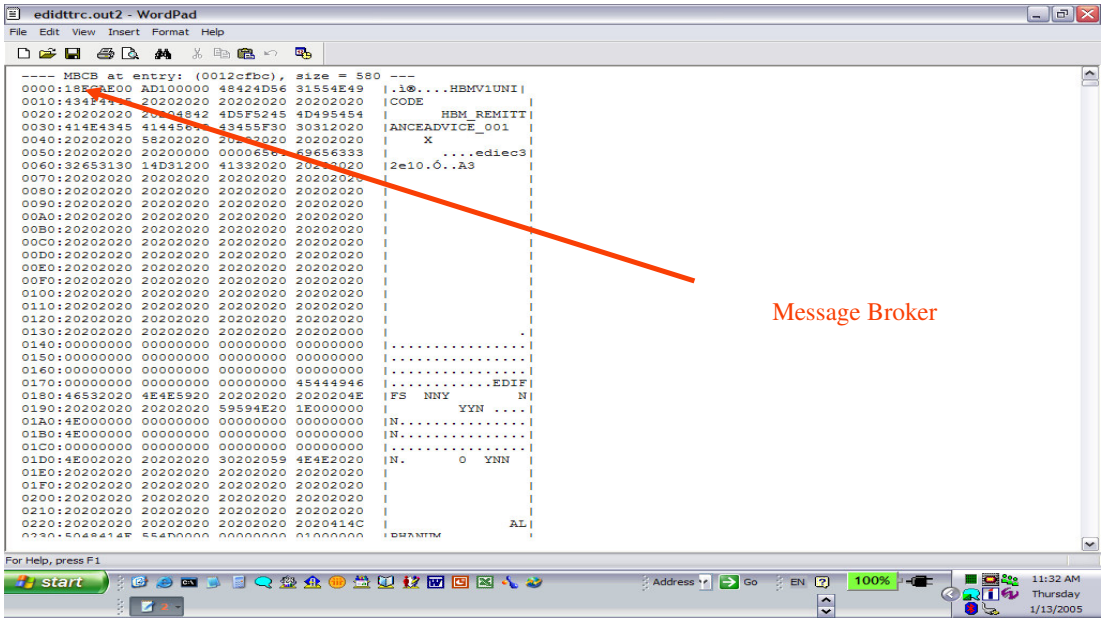
TRACELEVEL Keyword

- 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)

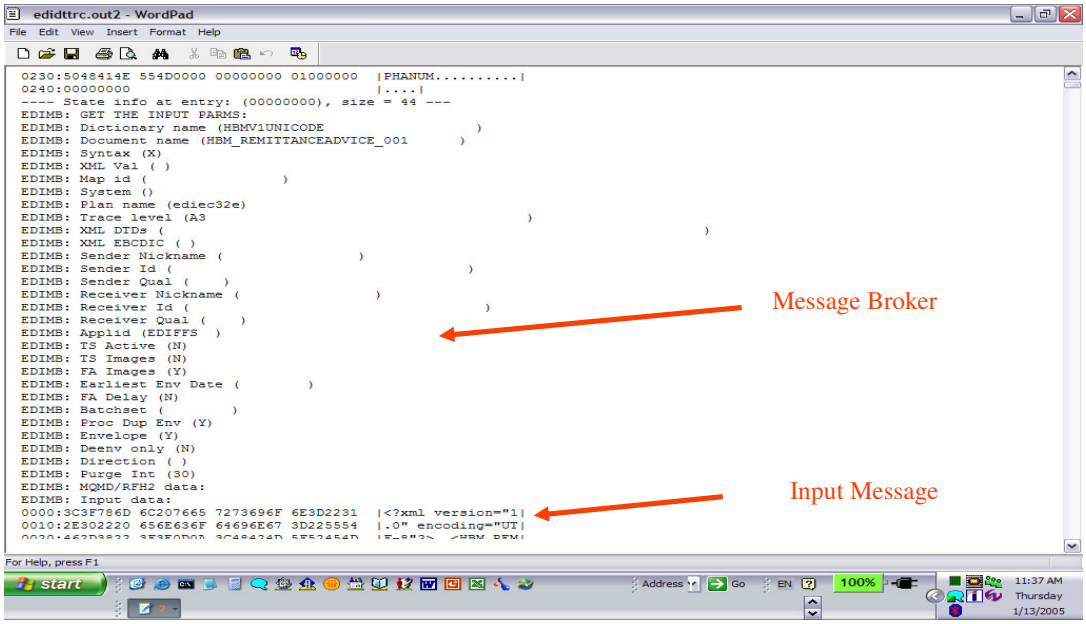
WDI development uses the trace for problem determination. There is a special trace called CheckPoint Trace. The CheckPoint trace shows date/time and on z/OS the heap storage used. This is for tracking of memory usage.

TRACE Example



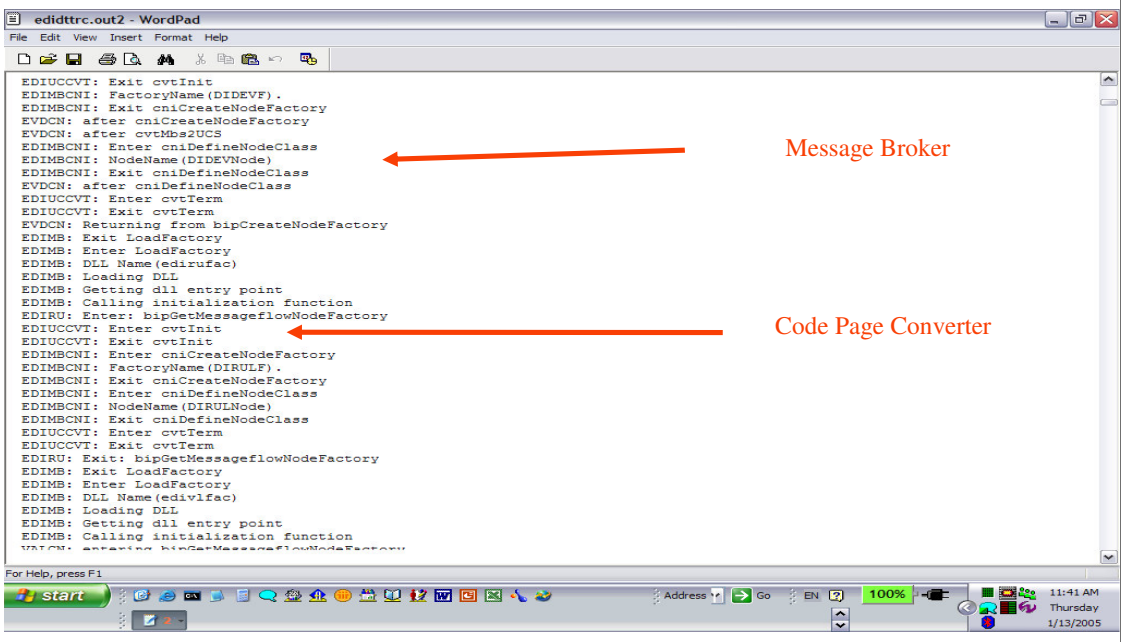
This is a section of the Data Transformation trace. The MBCB identifies the WDI Message Broker execution step.

TRACE Example

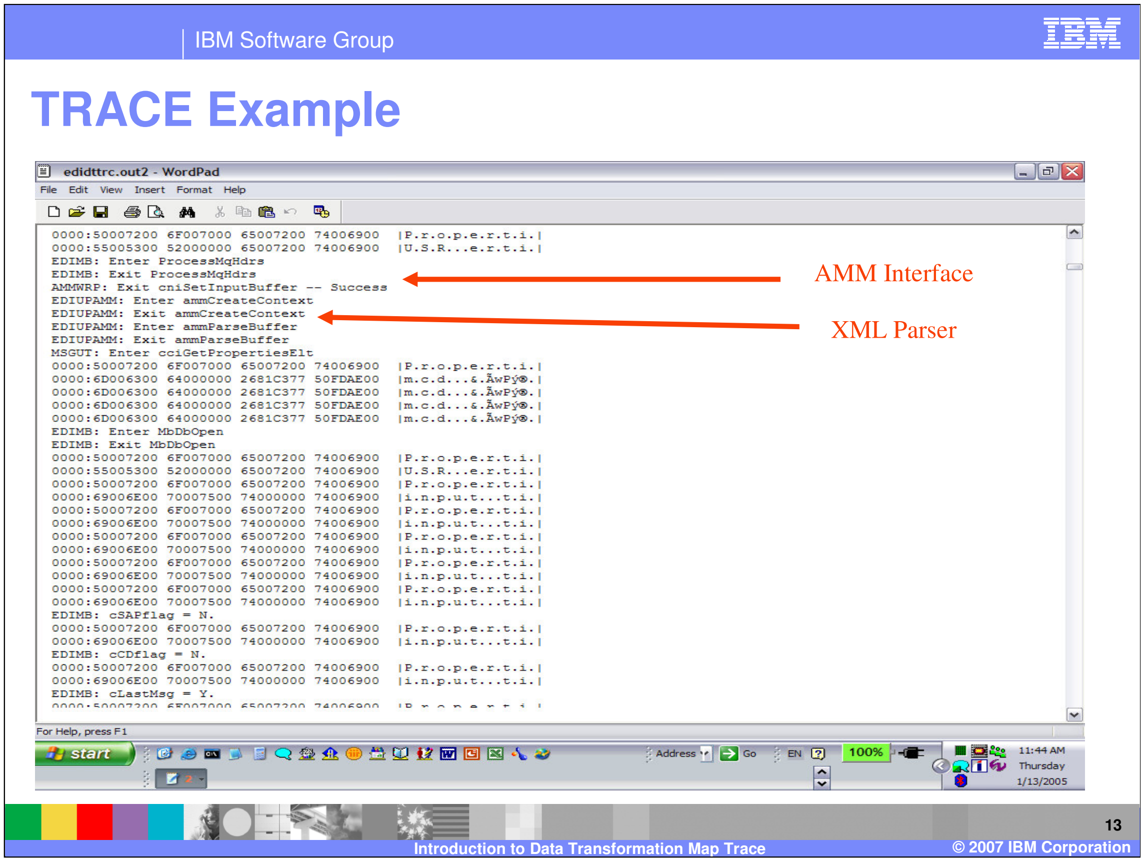


This section of the trace identifies the EDIMB: is one of the WDI Message Broker modules and also indicates WDI Message Broker execution. Input data: is the dump of the logical message that will be parsed.

TRACE Example

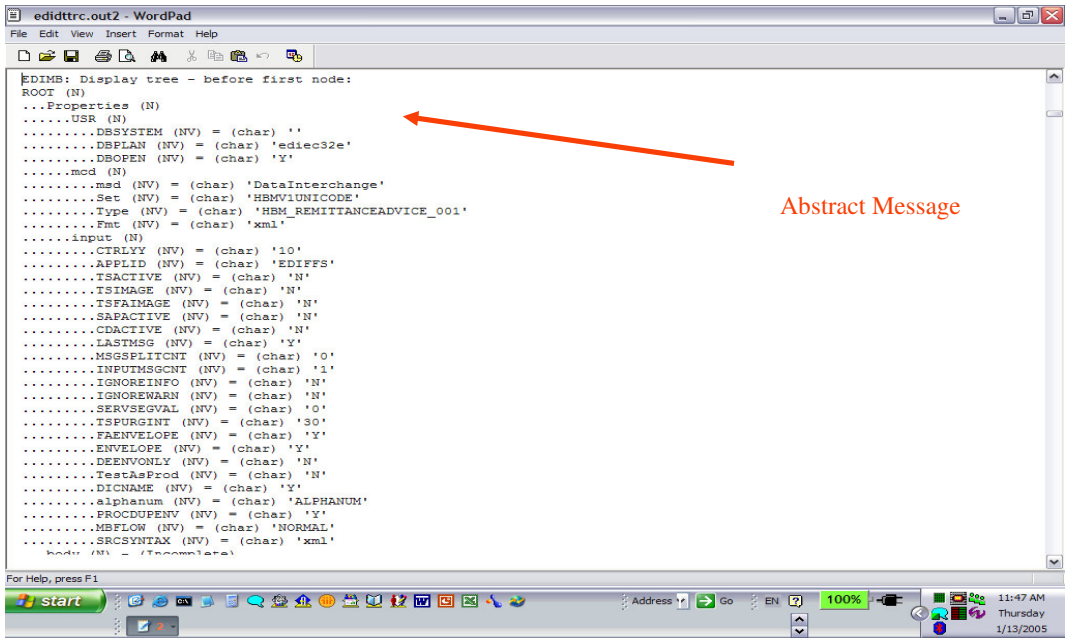


EDIMBCNI: is another WDI Message Broker module and also indicates WDI Message Broker execution. EDIUCCVT: is the code page converter module.



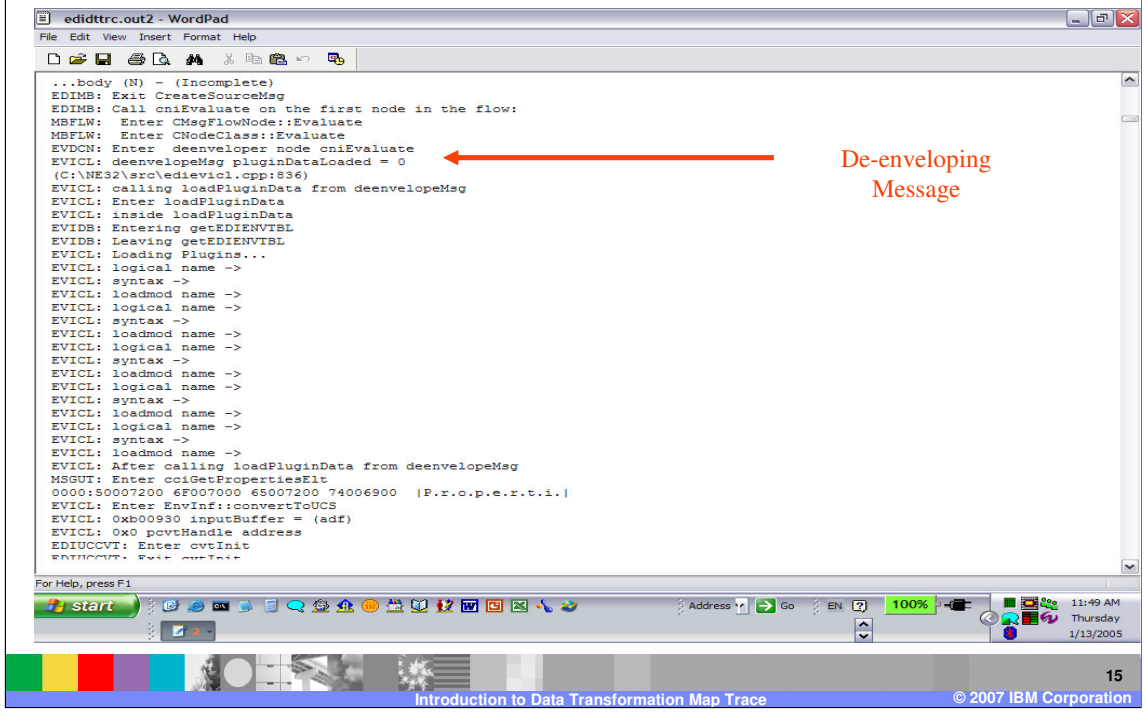
AMMWRP: is the WDI Abstract Message interface. EDIUPAMM: is the WDI XML parser interface.

TRACE Example



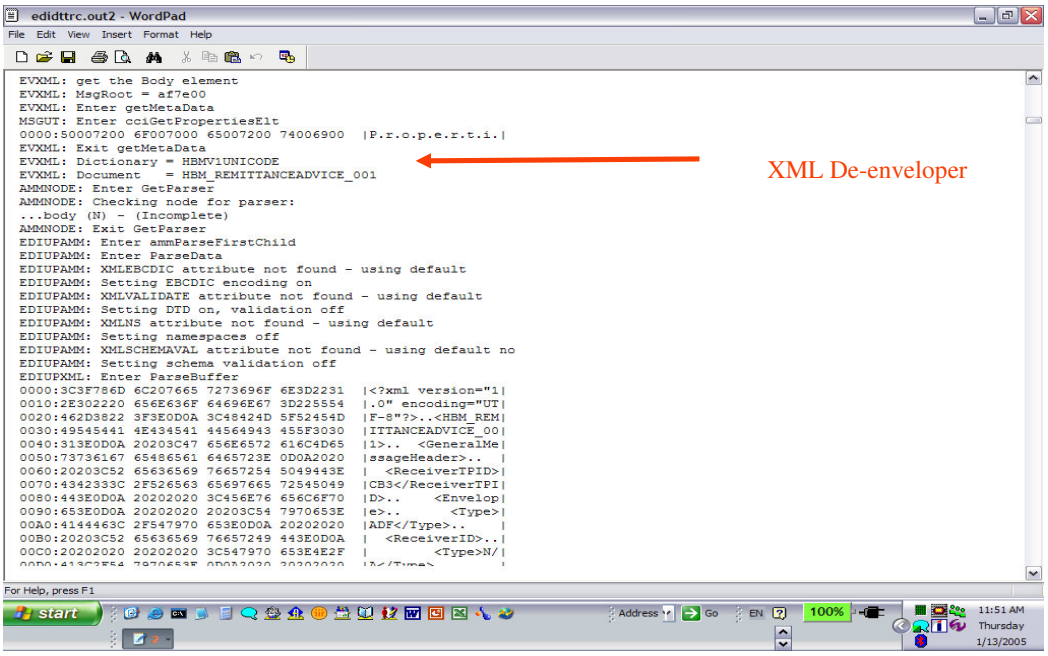
The EDIMB: Display tree is the display of the WDI Abstract Message.

TRACE Example



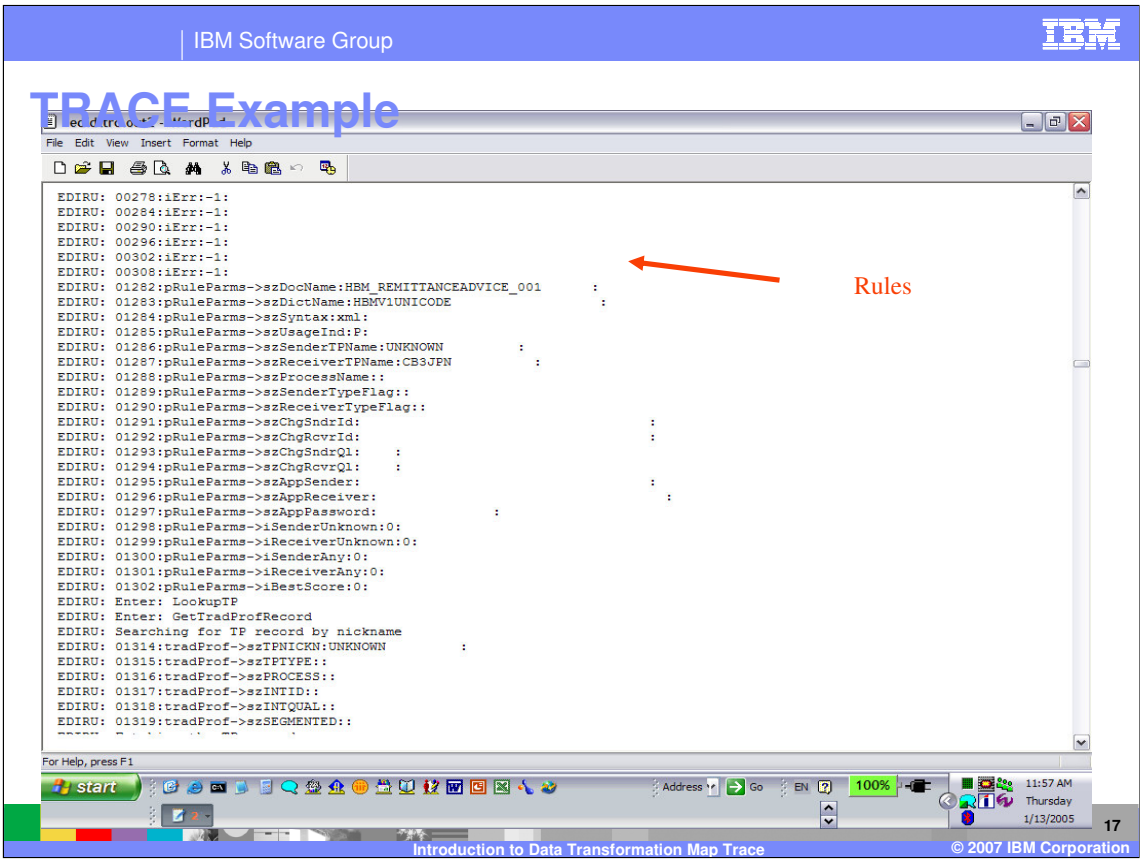
EVICL: indicates the deenvelope node execution.

TRACE Example



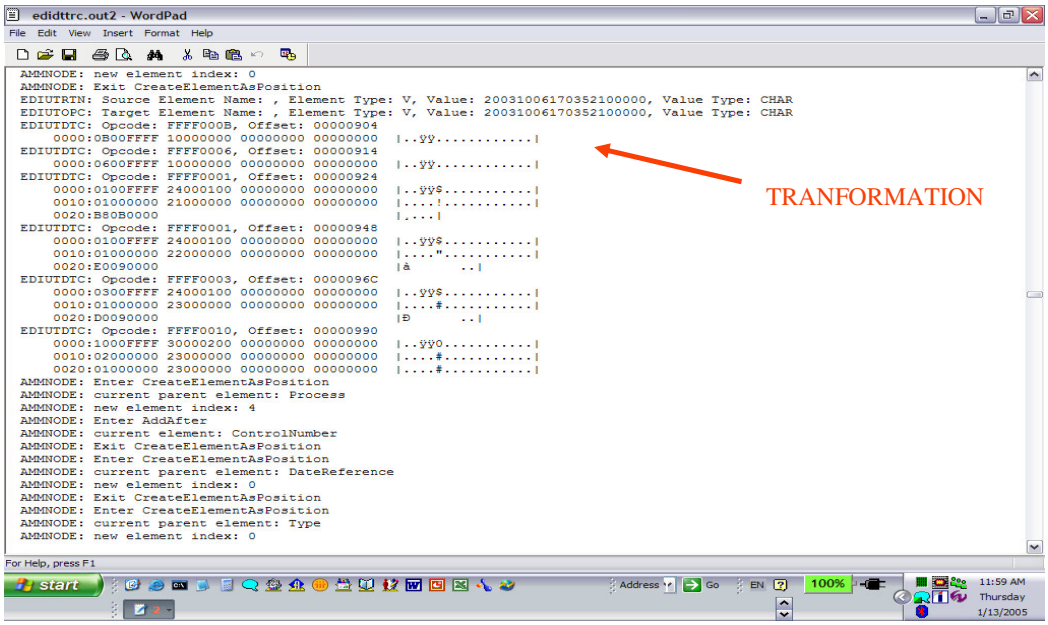
EVXML: identifies the deenveloper assigned is XML.

TRACE Example



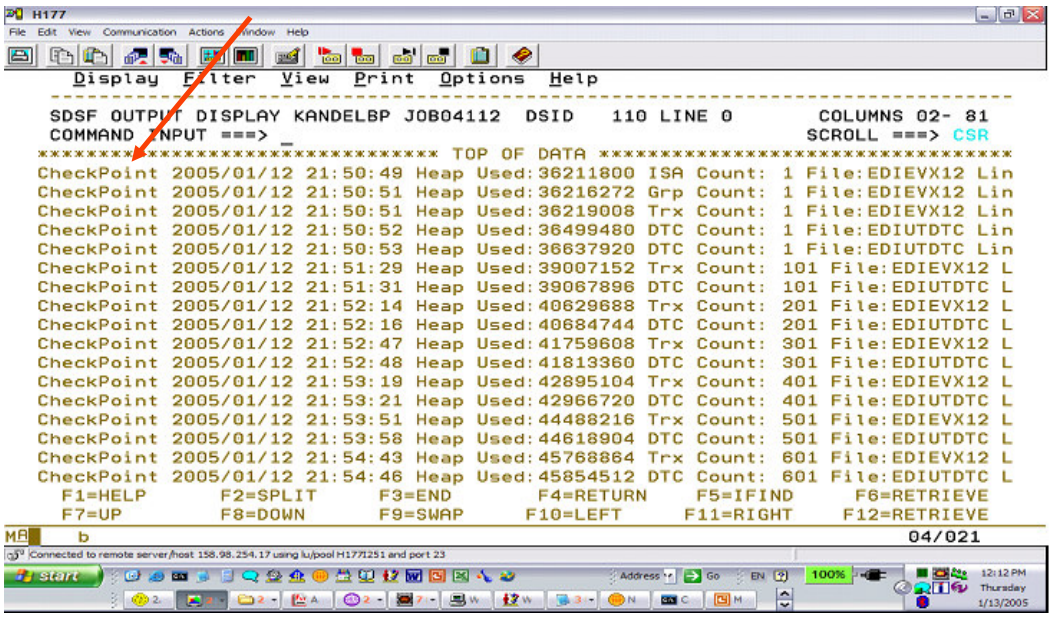
EDIRU: identifies the Rules node execution.

TRACE Example

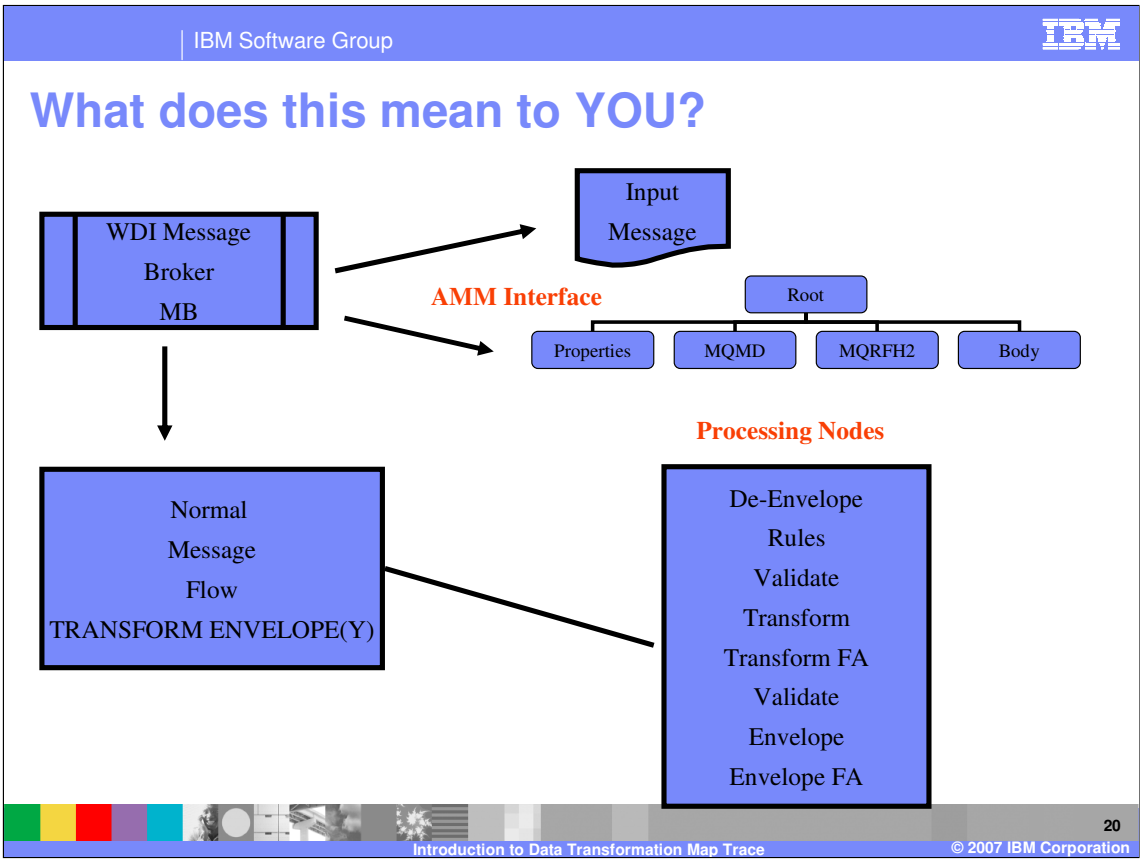


EDIUTRTN:, EDIUTOPC:, and EDIUTDTC: identify the Transform node execution.

TRACE Example



This is an example of the CheckPoint trace.



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	CICS	IMS	WMQ	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	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.

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.