

Displaying PI Single Message Performance Data of IE the Easy Way



Applies to:

SAP Process Integration (XI 3.0 to PI 7.1). For more information, visit the [Performance homepage](#).

Summary

A how-to for visualizing data of the runtime performance in the IE of PI with little overhead is presented. There are many powerful tools available in SAP Solution Manager and with the Solution Manager Diagnostics (SMD). For developers and first users of a system these tools are too often not available. In this Blog I will just show how to display this information with limited eyestrain. For this only a style sheet transformation and some extra java code is used. A standalone example is presented, but integration into PI is a possibility. Standard tools like XSLT and enhanced XSLT functions in java together with a XSLT processor are used. Download attached files [here](#).

Author: Frank Schmidt

Company: SAP America Inc.

Created on: 16 April 2011

Author Bio



Frank Schmidt works in the COE of AGS for SAP America Inc delivering assessments, supporting projects through escalations and standard AGS services.

Table of Contents

Introduction to the IE Performance Header	3
Solving the Presentation of Data Problem.....	4
Motivation for PI Developers:.....	4
Requirements for Solution:	4
The current standalone solution:.....	4
Components of the Solution:.....	4
The Solution:.....	4
Related Content.....	6
Copyright.....	7

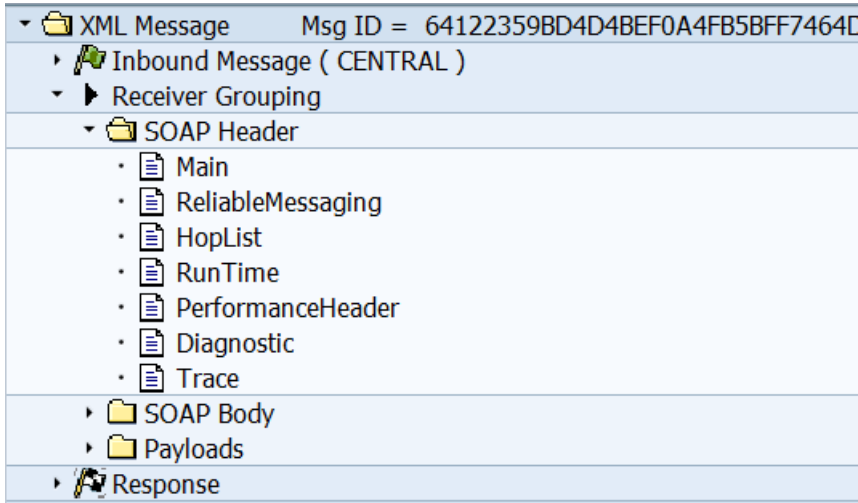
Introduction to the IE Performance Header

To collect data in the SOAP message as it passes through the Integration engine one has to be make sure that the IE configuration settings include the settings:

	PERF	MEASUREMENT_LEVEL	<input type="checkbox"/>	2
	PERF	MEASUREMENT_PERSIST	<input type="checkbox"/>	1

With the MEASUREMENT_LEVEL at least at 2 the SOAP message is collecting performance data in the <SOAP:HEADER> part of the XML that his the representation of the message as it flows through the Integration Engine (IE).

The SAP transaction 'SXMB_MONI' can be used to used to select messages for display and one can see the message status and the Header section of the message:



At this level everything looks clear and organized. The problem becomes apparent when one looks at the 'PerformanceHeader' part via the Standard SAP Transaction:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!-- Receiver Grouping -->
- <SAP:PerformanceHeader xmlns:SAP="http://sap.com/xi/XI/Message/30">
- <SAP:RunTimeItem>
  <SAP:Name
    type="ADAPTER_IN">INTEGRATION_ENGINE_HTTP_ENTRY</SAP:Name>
  <SAP:Timestamp type="begin"
    host="ncev0000012a">20110406153725.241</SAP:Timestamp>
</SAP:RunTimeItem>
- <SAP:RunTimeItem>
  <SAP:Name type="PARSER">PARSING_HTTP_ENTRY</SAP:Name>
  <SAP:Timestamp type="begin"
    host="ncev0000012a">20110406153725.241</SAP:Timestamp>
```

The timestamp is formatted in a not easily decipherable form. To make it more complicated the timestamp can be precise to the millisecond or the microsecond depending on the platform on which PI is running.

When people need to guess what information is presented they usually ignore it.

This readability challenge and the fact that I have to deal with PI performance data on a daily basis is the motivation for the presented alternative view of the data.

Solving the Presentation of Data Problem

XML is the coin in which most data is minted when it has to be shared with different receivers or when one wants to present it within a web browser easily. The specific namespace in which the PI SOAP message is presented only limits the legal data content and does not hinder the use of XSLT standard tools.

At a first glance this problem seems to be asking only for a simple transformation into an html document that can be looked at via any HTML 4.0 enabled web browser. To make the presentation more useful the execution times for each service should be displayed. After all this data should be used to document performance of message processed by the IE of PI.

Motivation for PI Developers:

Prove to your development lead that your mapping is performing well:

Check the execution time for service **PLSRV_MAPPING_REQUEST**.

Blame you system performance for any long delays:

Check the execution time for services **DB_ENTRY_QUEUING** and **DB_SPLITTER_QUEUING**.

Tip: Document performance in development, quality and production systems and enjoy life.

Note: The above only holds true if one is dealing with a very simple scenario.

Requirements for Solution:

- Present performance data of downloaded SOAP messages
- Hide complexity of different timestamp formats
- Show how long each pipeline service needed to execute

The current standalone solution:

The problem that is not solved via standard XSLT 1.0 here is the format of the timestamp. SAP is supporting many operating systems and the timestamp has a different length on MS Windows and the different UNIX flavors. The XSLT1.0 standard <http://www.w3.org/TR/xslt> is offering in section 14.2 a mechanism to create function extensions that are implemented by most XSLT processor implementations.

Components of the Solution:

- One XSL Stylesheet <http://www.w3.org/Style/XSL/>
- One JAVA jar file with the XSLT extension functions
- Small MS command.com script to execute via the open source APACHE XALAN XSLT <http://xml.apache.org/xalan-j/> processor.

The Solution:

There is more than one way to download the SOAP XML of a PI message. One can download the performance part only or the full message.

For the header only use the Stylesheet 'PIPERF.xsl' for the full SOAP message PISOAP.xsl.

There are many commercial and opensource XSLT processors available. For a standalone no cost solution you can download the binary distribution of APACHE XALAN via the above link.

You will have to adjust the Windows batch execution PI_Time.bat file to reflect the directory structure on your computer and the location of the PISOAP.jar that includes the java methods for the time conversion.

The command line will look like this:

Performance header only:

PI_Time.bat PIPerf1.xml PIPERF.xsl PIPerf1.html

Full SOAP message:

PI_Time.bat PISOAP1.xml PISOAP.xsl PISOAP1.html

An example of a generated html document is given below. Changes to formatting and look can be implemented easily with more effort spend on the presentation.

PI Performance extract from Integration Engine SOAP message:

Service Name:	Service Type:	Server Name:	Time Stamp:	Time Delta:
IDOC_ADAPTER	ADAPTER_IN	slqxq1	03/07/2011 19:25:16:207307	
IDOC_ADAPTER	ADAPTER_IN	slqxq1	03/07/2011 19:25:16:267372	60ms:65µs
INTEGRATION_ENGINE	CORE	slqxq1	03/07/2011 19:25:16:269298	
PLSRV_XML_VALIDATION_RQ_XMS	CORE	slqxq1	03/07/2011 19:25:16:273026	
PLSRV_XML_VALIDATION_RQ_XMS	CORE	slqxq1	03/07/2011 19:25:16:273066	40µs
INTEGRATION_ENGINE	CORE	slqxq1	03/07/2011 19:25:16:280934	11ms:636µs
DB_ENTRY_QUEUEING	DBQUEUE	slqxq1	03/07/2011 19:25:16:280969	
DB_ENTRY_QUEUEING	DBQUEUE	slqxq1	03/07/2011 19:25:20:145706	3s:864ms:737µs
PLSRV_XML_VALIDATION_RQ_INB	PLSRV	slqxq1	03/07/2011 19:25:20:152732	
PLSRV_XML_VALIDATION_RQ_INB	PLSRV	slqxq1	03/07/2011 19:25:20:181464	28ms:732µs
PLSRV_RECEIVER_DETERMINATION	PLSRV	slqxq1	03/07/2011 19:25:20:403476	
PLSRV_RECEIVER_DETERMINATION	PLSRV	slqxq1	03/07/2011 19:25:22:227794	1s:824ms:318µs
PLSRV_INTERFACE_DETERMINATION	PLSRV	slqxq1	03/07/2011 19:25:22:990404	
PLSRV_INTERFACE_DETERMINATION	PLSRV	slqxq1	03/07/2011 19:25:25:662970	2s:672ms:566µs
PLSRV_RECEIVER_MESSAGE_SPLIT	PLSRV	slqxq1	03/07/2011 19:25:25:890281	
PLSRV_RECEIVER_MESSAGE_SPLIT	PLSRV	slqxq1	03/07/2011 19:25:25:925181	34ms:900µs
DB_SPLITTER_QUEUEING	DBQUEUE	slqxq1	03/07/2011 19:25:25:925199	
DB_SPLITTER_QUEUEING	DBQUEUE	slqxq1	03/07/2011 19:25:27:056489	1s:131ms:290µs
PLSRV_MAPPING_REQUEST	PLSRV	slqxq1	03/07/2011 19:25:27:067015	
PLSRV_MAPPING_REQUEST	PLSRV	slqxq1	03/07/2011 19:25:27:577955	510ms:940µs
PLSRV_OUTBOUND_BINDING	PLSRV	slqxq1	03/07/2011 19:25:27:732611	
PLSRV_OUTBOUND_BINDING	PLSRV	slqxq1	03/07/2011 19:25:27:855575	122ms:964µs
PLSRV_XML_VALIDATION_RQ_OUT	PLSRV	slqxq1	03/07/2011 19:25:28:034583	
PLSRV_XML_VALIDATION_RQ_OUT	PLSRV	slqxq1	03/07/2011 19:25:28:058924	24ms:341µs
PLSRV_CALL_ADAPTER	PLSRV	slqxq1	03/07/2011 19:25:28:241271	
PLSRV_CALL_ADAPTER	PLSRV	slqxq1	03/07/2011 19:25:29:030492	789ms:221µs

Related Content

[Process Integration - Exchange Infrastructure](#)

[End-to-End Process Integration How-to Guides for SAP NetWeaver 7.0](#)

[SAP Blogs - SAP Solution Manager](#)

Download attached files [here](#)

For more information, visit the [Performance homepage](#)

Copyright

© Copyright 2011 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.