

Tracing Solutions in BizTalk

#AimsPerformancePro Webinar

Ahmed Taha

Nice to meet you!

Technical Architect at Link Development, Cairo, Egypt

Integration enthusiast

Avid Reader 😊

AimsPerformancePro



AIMS
1,252 followers
8mo

We're happy to welcome [Ahmed Taha](#) as our newest [#aimsperformancepro](#)! Check out this blog post to read about Ahmed's first impressions installing and using AIMS: <https://lnkd.in/dx8fiZQ>



First experiences with AIMS from a Microsoft integration expert
aims.ai



#AimsPerformancePro

AimsPerformancePro



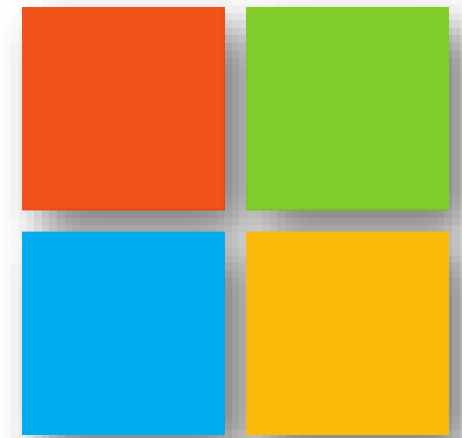
#AimsPerformancePro



BizTalk or Azure Integration Services?



“Our Integration Product Roadmap”



“One thing has prevailed over these many years, existing enterprise customers running business critical workloads using BizTalk Server on-premises - and Microsoft’s continued commitment and support to these customers.” - *Jon Fancey, Azure Integration Services PM Lead*

https://blogs.msdn.microsoft.com/biztalk_server_team_blog/2018/08/09/our-integration-product-roadmap/

Goals for tracing BizTalk solutions

Raises the end-to-end visibility of the integration technical solution

Assists Developers in understanding the behavior of the solution

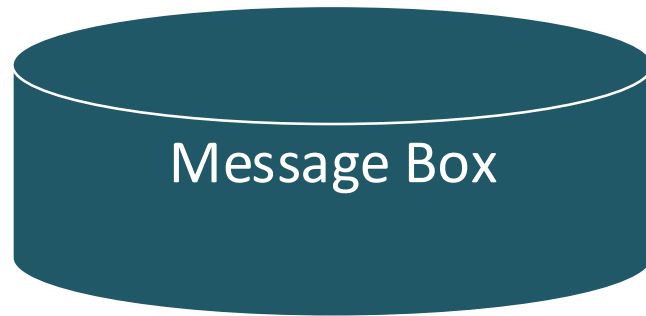
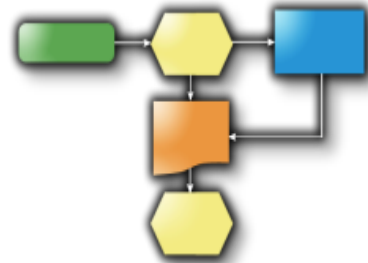
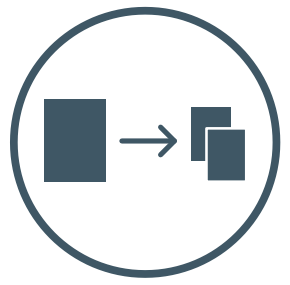
Provides detailed enough trace for troubleshooting issues

Provides clear visibility, minimizes the noise

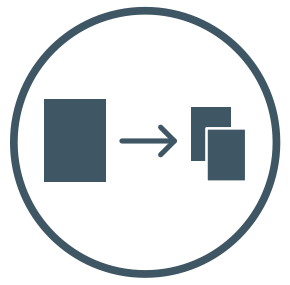
Has minimal performance impact on the solution

Common approaches to troubleshoot BizTalk Solutions

File-based



File-based

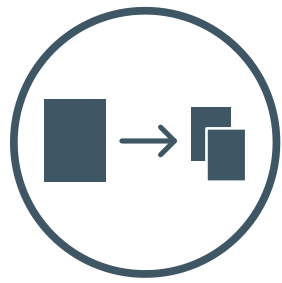


Conducts testing at the edges of the integration solution

Troubleshoots runtime issues end-to-end

Provides temporary troubleshooting – unenlist file port when not needed

File-based common pitfalls



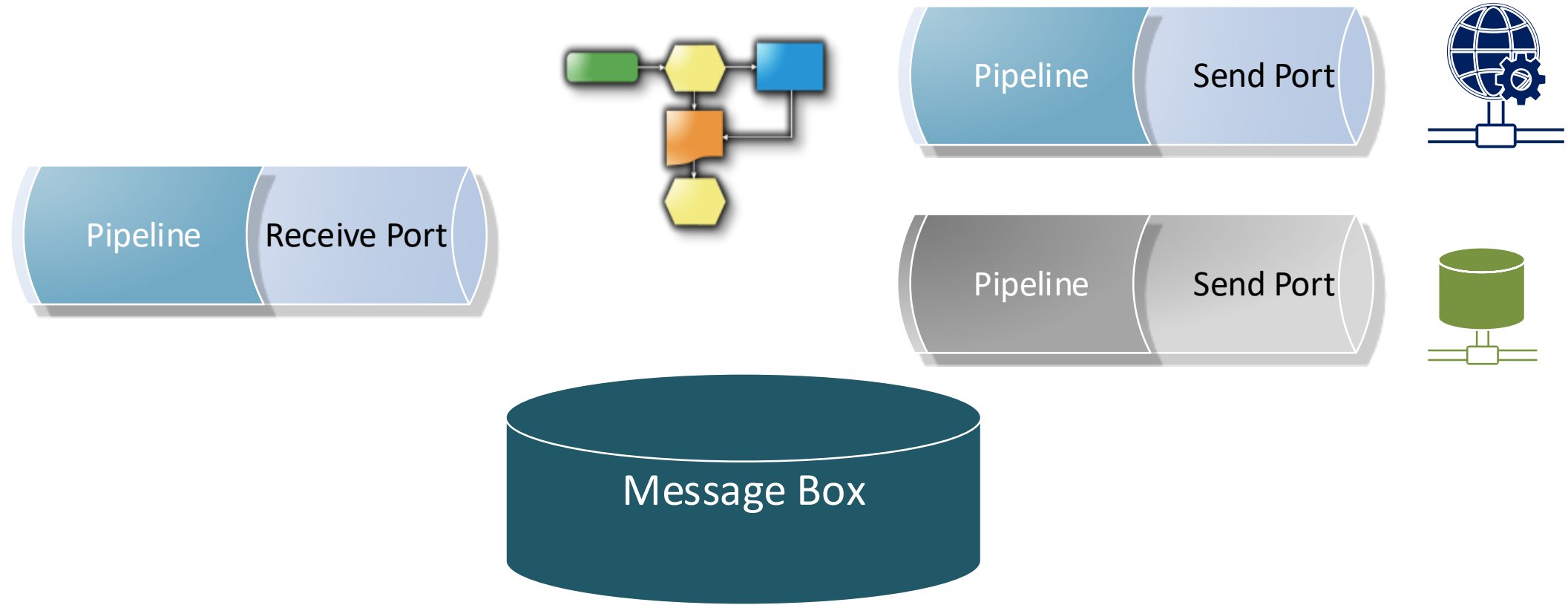
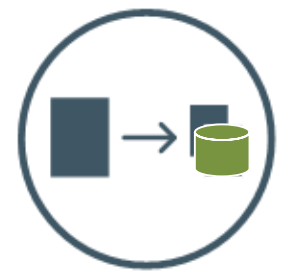
Adds more load on the BizTalk engine while being used

Requires a cleanup process for the folder being use

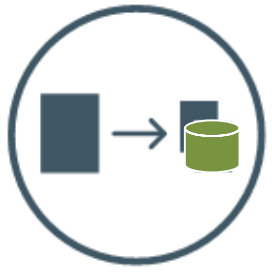
Requires proper governance over folder permissions

Could expose confidential, critical information in the message payload

Custom DB



Custom DB

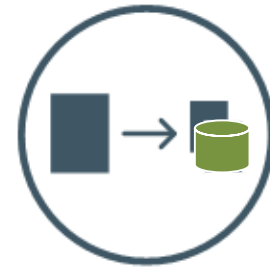


Traced data is stored in a structured format

Can conduct queries against the Traced data

Selective Message payload properties can be traced

Custom DB common pitfalls



Adds more load on the BizTalk engine while being used

Custom DB will need to be managed and maintained

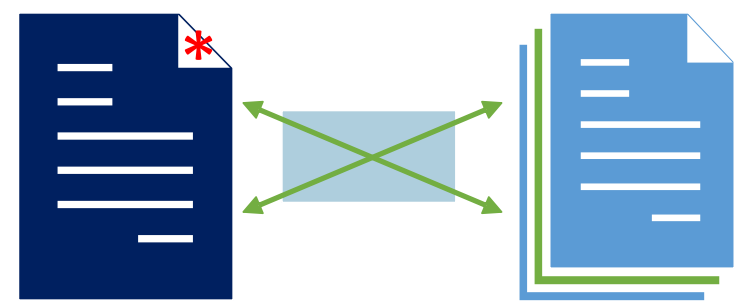
Custom DB security need to be governed

Data retention and archiving techniques will be required

Sometimes, customers mingle business reports within this custom DB

Can't be used to accurately measure performance of a process

Test Maps with Live messages*



A screenshot of the Microsoft BizTalk Studio interface. The main window shows a map editor for 'CustomerV2.xsd'. The map has several input and output ports connected by lines. A context menu is open over the map, with the 'Test Map' option highlighted in yellow. The menu also includes options like 'Open', 'Validate Map', 'Debug Map', etc. The Solution Explorer on the right shows a project structure with files like 'CustomerV1.xsd', 'CustomerV2.xsd', and 'Service1.BindingInfo.xml'. The bottom right corner shows a preview of XML data.

Administration Console



BizTalk Server Administration Console

File Action View Help

Console Root

- BizTalk Server Administration
 - BizTalk Group [WIN-URNP6CC6A75:Bi
 - Applications
 - Parties
 - Platform Settings
 - Hosts
 - Host Instances
 - Servers
 - Message Boxes
 - Adapters
 - Event Viewer (Local)

Group Overview

Refresh this page to update results (press F5)

Group Hub New Query

Configuration Overview

Group name: BizTalk Group	Applications (5)	▶
Server: WIN-URNP6CC6A75	Host Instances (1)	✓
Database: BizTalkMgmtDb	Adapter Handlers (49)	✓

[Learn about Administration and Troubleshooting](#)

Work in Progress

Running service instances	0
- Dehydrated orchestrations	0
- Retrying and idle ports	0
- Ready service instances	0
- Scheduled service instances	0

Suspended Items

Suspended service instances	0
- Resumable	0
- Non-resumable	0

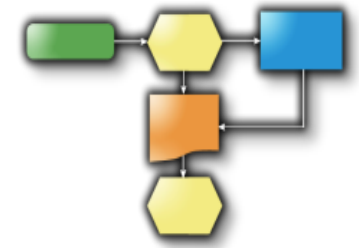
Grouped Suspended Service Instances

Grouped by Application	Grouped by Service Name
None	None
Grouped by Error Code	Grouped by URI
None	None

Tracked Service Instances

Tracked service instances	Tracked message events
- Completed instances	- Transmission failure events

Orchestration Debugger



Generic [completed] - Orchestration Debugger [replay] - {2C8457F9-54A0-46E4-9BE0-48B7...}

File Debug Help

Tracked Events		
	Action Name	Action Type
1	Initialization	Orchestra
2	Receive_837	Receive
3	Receive_837	Receive
4	ConstructMess...	Construct
5	ConstructMess...	Construct
6	ConstructMess...	Construct
7	ConstructMess...	Construct
8	Archive_Data	VariableA
9	Archive_Data	VariableA
10	Send_Output	Send
11	Send_Output	Send
12	Initialization	Orchestra

Orchestration

```
graph TD; Start(( )) --> Receive_837[Receive_837]; Receive_837 --> subgraph ConstructMessage_ECSIFXML; Transform_E_[Transform_E_]; end; ConstructMessage_ECSIFXML --> ConstructMessage_ECISIFFF[ConstructMessage_ECISIFFF];
```



Visual Studio good old fashioned Debugging

SampleTh - Microsoft Visual Studio (Administrator)

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP

Start Debug

Attach to Process

Transport: Default

Qualifier: WIN-URNP6CC6A75 Find...

Transport Information
The default transport lets you select processes on this computer or a remote computer running the Microsoft Visual Studio Remote Debugging Monitor (MSVSMON.EXE).

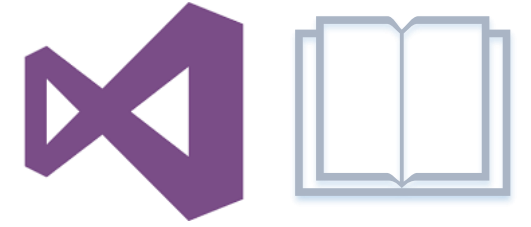
Attach to: Automatic: Managed (v4.5, v4.0) code Select...

Process	ID	Title	Type	User Name	Session
BTSNTSvc.exe	6056		Managed (v4....	WIN-URNP6CC6A75...	0
cmd.exe	6340	Administrator: Developer Command Prompt fo...	x64	WIN-URNP6CC6A75...	1
conhost.exe	4500		x64	WIN-URNP6CC6A75...	1
conhost.exe	4872		x64	WIN-URNP6CC6A75...	1
conhost.exe	6984		x64	WIN-URNP6CC6A75...	1
conhost.exe	3312		x64	WIN-URNP6CC6A75...	1
conhost.exe	5832		x64	WIN-URNP6CC6A75...	1
conhost.exe	3728		x64	WIN-URNP6CC6A75...	1
devenv.exe	2636	Best Practices for Instrumenting High Performa...	Managed (v4....	WIN-URNP6CC6A75...	1
devenv.exe	5844	BTSDEMO - Microsoft Visual Studio (Administra...	Managed (v4....	WIN-URNP6CC6A75...	1

Show processes from all users Refresh

Attach Cancel

.NET Logging Frameworks



Log4net

Enterprise Library

Event Log – Log Application State per Server



The screenshot shows the Windows Event Viewer interface. The left-hand pane displays the 'Event Viewer (Local)' tree with 'Windows Logs' expanded to 'Application'. The main pane shows a list of 35 events. The selected event is an error from 'CustomerBTS' at 12/29/2018 12:47:59 PM. Below the list, the details for 'Event 0, CustomerBTS' are shown, including a 'General' tab with the message: 'Customer Orchestration encountered an Error'.

Level	Date and Time	Source	Event ID	Task Category
Information	1/11/2019 1:20:13 PM	Security-SPP	900	None
Error	12/29/2018 1:05:55 PM	CustomerBTS	0	None
Error	12/29/2018 1:05:55 PM	BizTalk Server	5754	BizTalk Server
Error	12/29/2018 12:47:59 PM	CustomerBTS	0	None
Error	12/29/2018 12:47:58 PM	BizTalk Server	5754	BizTalk Server
Information	12/29/2018 12:29:24 PM	BizTalk Server	5410	BizTalk Server
Information	12/29/2018 12:29:22 PM	BizTalk Server	5410	BizTalk Server

Event 0, CustomerBTS

General Details

Customer Orchestration encountered an Error

ESB Portal – Centrally Log Application State

The screenshot displays the ESB Management Console interface. At the top, there is a navigation bar with tabs for Home, Faults, Alerts, Reports, Registry, Admin, and My Settings. Below this, a breadcrumb trail shows 'Home > Faults'. A red warning icon is followed by the heading 'Faults'. Underneath, there is a 'Filter Results' section with various input fields: 'Application' (set to 'All Applications'), 'Retrieve records within last' (set to 'day'), 'Records Per Page' (set to '20'), 'Fault Code', 'Fault Category', 'Error Type', 'Min. Fault Severity (1-4)', 'Max. Fault Severity (1-4)', and 'Submission Status' (unchecked). There is an 'Apply Filters' button and a 'Replay Messages' button. Below the filters, there is an 'Export to Excel' link. The main content area is a table with the following columns: Severity, Date, Code, Category, Application, Error Type, Scope, and Service Name. One fault record is visible, marked as 'Critical'.

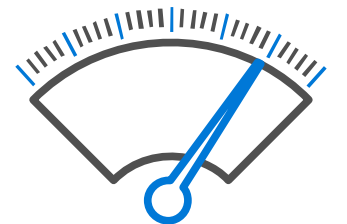
Severity	Date	Code	Category	Application	Error Type	Scope	Service Name
Critical	30/11/2011 13:28:17		Process Questions	Madurai.BizTalk.ReplayMessageSample	FormatException	Construct_msgFaultMessage	Madurai.BizTalk.ReplayMessageSample.Proc...

CAT Instrumentation Framework



Provide ultra-high performance Tracing for BizTalk solutions different artifacts

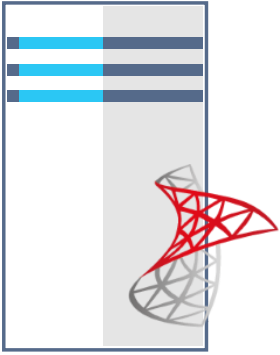
Custom pipeline components, Maps, Orchestrations, Custom Code, BRE, BAM



“The BizTalk Solution Instrumentation Framework is the recommended way to provide tracing for ultra-high performance scenarios as it uses kernel-level features of Event Tracing for Windows (ETW)” - *Dan Rosanova - Microsoft BizTalk Server 2010 Patterns*

Low performance overhead, suitable for tracing on production servers



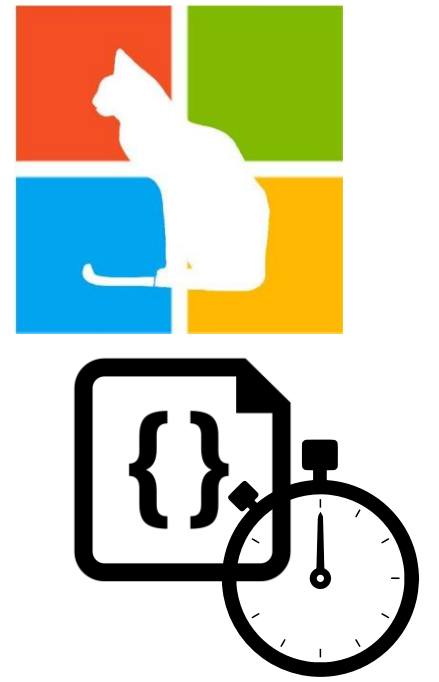


Experience in Low latency Solution



#AimsPerformancePro

Can be used to measure components
scope latency



Tracing output filtration based on the artifact type



Events can be traced as information,
warning, error trace



Tracing can be captured Synchronously
using Debug View



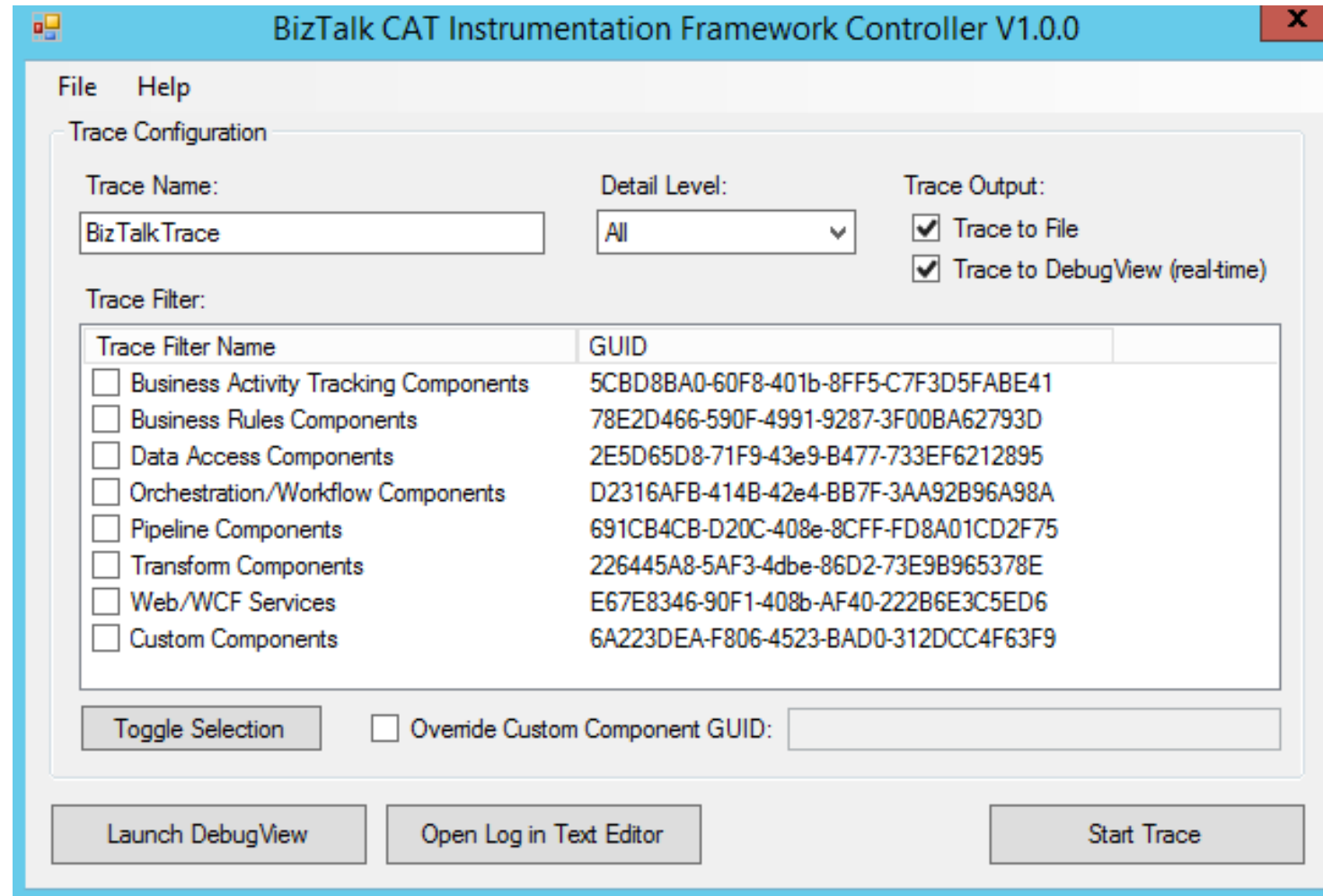
Tracing can be captured Asynchronously to a text file



CAT Instrumentation Framework (CAT IF)



CAT Instrumentation Controller add-on





Demo



CAT IF Tips

Text File – Async - log data is viewable in a text file only after a trace is stopped

Debug View – Real-time tracing, with filter options

Multiple Servers, will be traced separately

Add standardized naming conventions/error codes to your logs

Installation prerequisites:

You will need to compile the project with Framework 4.0

Add a strong name key to the project

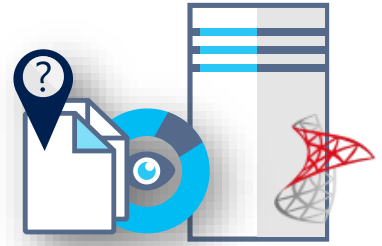
GAC Microsoft.BizTalk.CAT.BestPractices.Framework.dll

Add Reference to Microsoft.BizTalk.CAT.BestPractices.Framework.dll



Too much trace is not good too!

Recap



There is no right or wrong technology in the absolute sense

BizTalk Solutions should leverage CAT IF, for a tracing and debugging purposes

Use CAT IF Controller add-on for GUI control

Find the right level for tracing information, enough to raise the solution visibility

Logging Application states is different from tracing the application

Operational and Business Monitoring should leverage the appropriate tools

BizTalk CAT IF Resources

BizTalk CAT Instrumentation Framework

<https://github.com/tfabraham/BizTalkCATIFController>

BizTalk CAT Instrumentation Framework Controller

https://github.com/tfabraham/BizTalkCATIFController/blob/master/tools/BizTalkCATInstrumentationFrameworkV1_4.zip

Instrumentation Best Practices for High Performance BizTalk Solutions

<https://docs.microsoft.com/en-us/biztalk/technical-guides/instrumentation-best-practices-for-high-performance-biztalk-solutions>

BizTalk Server 2013 R2: Instrumenting an orchestration with ETW

<https://social.technet.microsoft.com/wiki/contents/articles/30523.biztalk-server-2013-r2-instrumenting-an-orchestration-with-etw.aspx>

Recommended Books



Professional Expertise Distilled

Microsoft BizTalk Server 2010 Patterns

Create effective, scalable solutions with Microsoft BizTalk Server 2010

Dan Rosanova

[PACKT] enterprise 
PUBLISHING



Professional Expertise Distilled

SOA Patterns with BizTalk Server 2013 and Microsoft Azure *Second Edition*

Learn how to create and implement SOA strategies on the Microsoft technology stack using BizTalk Server 2013 and Azure Integration platforms

Richard Seroter
Johann Cooper
Mahindra Morar

Mark Brimble
Colin Dijkstra

[PACKT] enterprise 
PUBLISHING

Q&A

ahmedmmt@outlook.com

Thank you