



Developer Best Practices:

Troubleshooting & Debugging

Sergey Marenich

Retail-Commerce Edition Team Lead / Architect

smarenich@acumatica.com | acumatica.com/developers

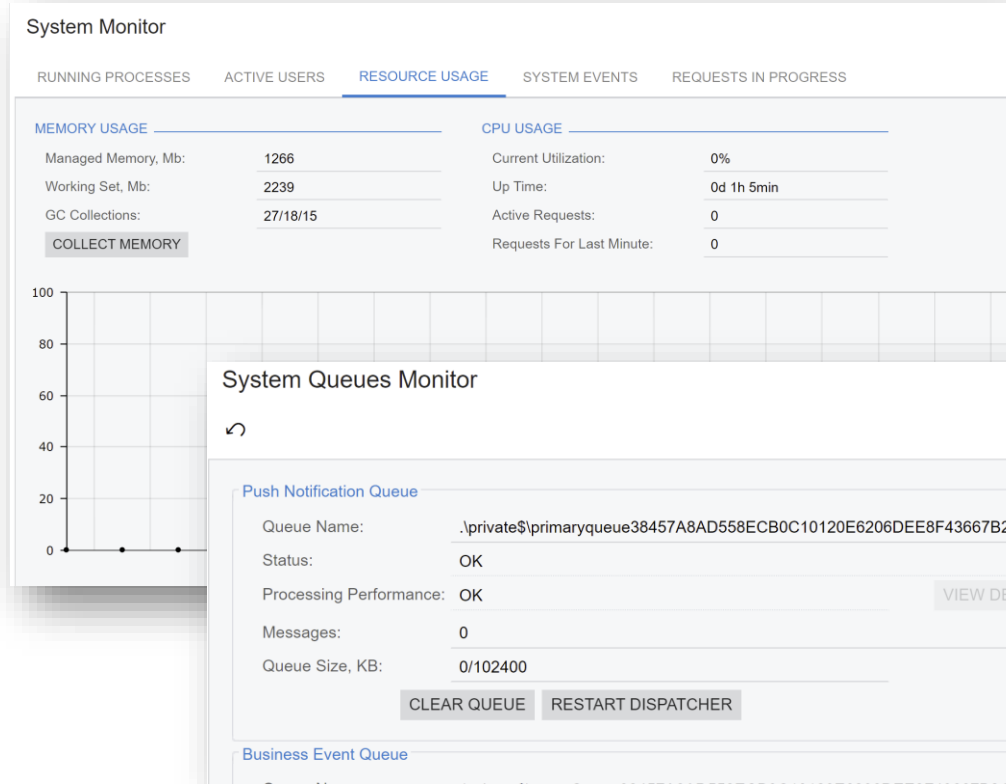


The Best Debug is No Debug: Tools

Acumatica System Monitors

Used to investigate system performance:

- CPU & Memory Load
- Running Background Processes
- Active Users
- System Events
- Queue Loads



Request Profiler

Used to:

- Understand user's flow
- View SQL queries
- Track history of errors
- Trace heavy actions

Request Profiler

REFRESH RESULTS CLEAR LOG EXPORT IMPORT

☒ Default Logging (Expensive Requests and Requests with Important Exceptions)

REQUEST LOGGING






☐ Log Requests (Apply Filter) Server Time Threshold: URL:
SQL Count Threshold: Username:



SQL LOGGING

☐ Log SQL (Apply Filter) Row Count Threshold: Executed by:
SQL Time Threshold: ☐ Include C

REQUESTS

 SQL EXCEPTIONS EVENT LOG

   VIEW SQL VIEW EVENT LOG OPEN URL PIN/UNPIN  

		Request Start Time	Username	URL	Screen	Request Type	Sta	Com	Target
>		01 Mar 17:54:2	admin	~/pages/sm/...	DB000015	Screen			ctl00



The Best Debug is No Debug: Logs

To Log All Exceptions in the System to a File

In web.config file:

```
<appSettings>
```

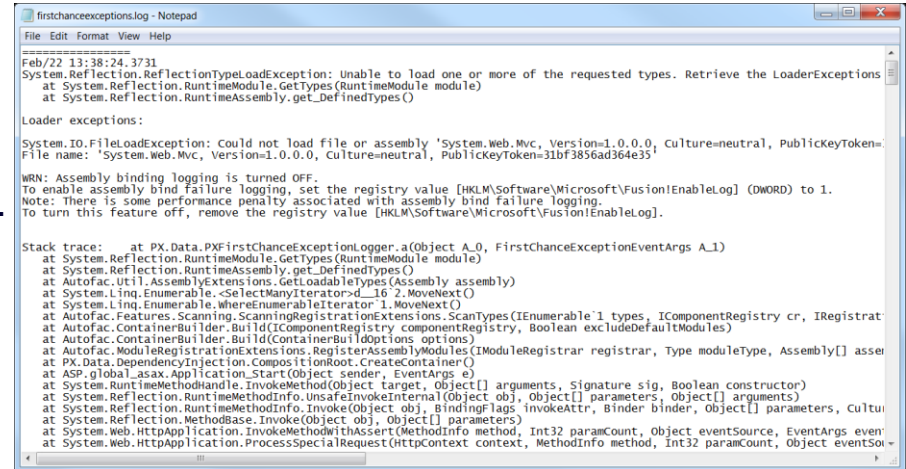
```
  <add key="EnableFirstChanceExceptionsLogging" value="true" />
```

```
  <add key="FirstChanceExceptionsLogFileName" value="mylog.log" />
```

```
</appSettings>
```

Note:

By default, the first-chance exception log file is saved in the <Site>\App_Data folder of the website.



```
firstchanceexceptions.log - Notepad
File Edit Format View Help
=====
Feb/22 13:38:24.3731
System.Reflection.ReflectionTypeLoadException: Unable to load one or more of the requested types. Retrieve the LoaderExceptions
  at System.Reflection.RuntimeModule.GetTypes(RuntimeModule module)
  at System.Reflection.RuntimeAssembly.get_DefinedTypes()
Loader exceptions:
System.IO.FileLoadException: Could not load file or assembly 'System.Web.Mvc, Version=1.0.0.0, Culture=neutral, PublicKeyToken=
File name: 'System.Web.Mvc, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e33'
WRN: Assembly binding logging is turned OFF.
To enable assembly bind failure logging, set the registry value [HKLM\Software\Microsoft\Fusion!EnableLog] (DWORD) to 1.
Note: There is some performance penalty associated with assembly bind failure logging.
To turn this feature off, remove the registry value [HKLM\Software\Microsoft\Fusion!EnableLog].

Stack trace:
   at PX.Data.PXFirstChanceExceptionLogger.a(Object A_0, FirstChanceExceptionEventArgs A_1)
   at System.Reflection.RuntimeModule.GetTypes(RuntimeModule module)
   at System.Reflection.RuntimeAssembly.get_DefinedTypes()
   at Autofac.Util.AssemblyExtensions.GetLoadableTypes(Assembly assembly)
   at System.Linq.Enumerable.<SelectManyIterator>d__16.MoveNext()
   at System.Linq.Enumerable.WhereEnumerableIterator.MoveNext()
   at Autofac.Features.Scanning.ScanningRegistrationExtensions.ScanTypes(IEnumerable`1 types, IComponentRegistry cr, IRegistrar
   at Autofac.ContainerBuilder.Build(IComponentRegistry componentRegistry, Boolean excludeDefaultModules)
   at Autofac.ContainerBuilder.Build(ContainerBuildOptions options)
   at Autofac.ModuleRegistrationExtensions.RegisterAssemblyModules(IModuleRegistrar registrar, Type moduleType, Assembly[] asse
   at PX.Data.DependencyInjection.CompositionRoot.CreateContainer()
   at ASP.global_asax.Application.Start(Object sender, EventArgs e)
   at System.RuntimeMethodHandle.InvokeMethod(Object target, Object[] arguments, Signature sig, Boolean constructor)
   at System.Reflection.RuntimeMethodInfo.UnsafeInvokeInternal(Object obj, Object[] parameters, Object[] arguments)
   at System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, Cultu
   at System.Reflection.MethodBase.Invoke(Object obj, Object[] parameters)
   at System.Web.HttpApplication.InvokeMethodWithAssert(MethodInfo method, Int32 paramCount, Object eventSource, EventArgs even
   at System.Web.HttpApplication.ProcessSpecialRequest(HttpContext context, MethodInfo method, Int32 paramCount, Object eventSou=
```

In-Browser Customization Code + PXTrace

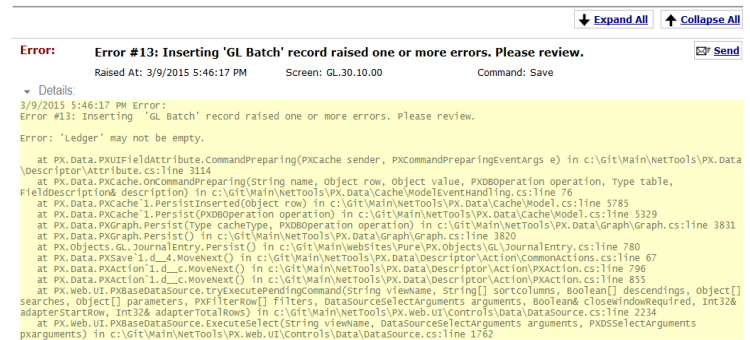
Do you have a complicated case on the customer instance?

Publication of the customization code written in the Browser will not restart Application Domain, so you can inject some methods and write diagnostics to PXTrace:

- `PX.Data.PXTrace.WriteError(string message)`
- `PX.Data.PXTrace.WriteWarning(string message)`
- `PX.Data.PXTrace.WriteInformation(string message)`

Same for the Code Behind Files (*.aspx.cs)

Acumatica Trace:



The screenshot shows the Acumatica Trace interface. At the top, there are buttons for 'Expand All' and 'Collapse All'. Below that, an error message is displayed: 'Error: Error #13: Inserting 'GL Batch' record raised one or more errors. Please review.' The error was raised on 3/9/2015 at 5:46:17 PM, with the screen 'GL.30.10.00' and the command 'Save'. A 'Send' button is also present. Below the error message, there is a 'Details' section that is expanded, showing the stack trace. The stack trace starts with 'Error: 'Ledger' may not be empty.' and continues with several lines of code from the PX.Data.PXCache and PX.Data.PXGraph classes, indicating the sequence of operations that led to the error.

```
Error: Error #13: Inserting 'GL Batch' record raised one or more errors. Please review.
Raised At: 3/9/2015 5:46:17 PM      Screen: GL.30.10.00      Command: Save

Details:
3/9/2015 5:46:17 PM Error:
Error #13: Inserting 'GL Batch' record raised one or more errors. Please review.

Error: 'Ledger' may not be empty.
at PX.Data.PXUIFieldAttribute.CommandPreparing(PXCache sender, PXCommandPreparingEventArgs e) in c:\Git\Main\NetTools\PX.Data\Descriptor\Attribute.cs:line 3114
at PX.Data.PXCache.OnCommandPreparing(String name, Object row, Object value, PXBOperation operation, Type table, FieldDescription& description) in c:\Git\Main\NetTools\PX.Data\Cache\Model\EventHandling.cs:line 76
at PX.Data.PXCache`1.PersistInserted(Object row) in c:\Git\Main\NetTools\PX.Data\Cache\Model.cs:line 5785
at PX.Data.PXCache`1.Persist(PXBOperation operation) in c:\Git\Main\NetTools\PX.Data\Cache\Model.cs:line 5329
at PX.Data.PXGraph.Persist(Type cacheType, PXBOperation operation) in c:\Git\Main\NetTools\PX.Data\Graph\Graph.cs:line 3831
at PX.Data.PXGraph.Persist() in c:\Git\Main\NetTools\PX.Data\Graph\Graph.cs:line 3820
at PX.Objects.GL.JournalEntry.Persist() in c:\Git\Main\NetTools\PX.Objects\GL\JournalEntry.cs:line 780
at PX.Data.PXSave`1.d__4.MoveNext() in c:\Git\Main\NetTools\PX.Data\Descriptor\Action\CommonActions.cs:line 67
at PX.Data.PXAction`1.d__c.MoveNext() in c:\Git\Main\NetTools\PX.Data\Descriptor\Action\PXAction.cs:line 796
at PX.Data.PXAction`1.d__c.MoveNext() in c:\Git\Main\NetTools\PX.Data\Descriptor\Action\PXAction.cs:line 855
at PX.Web.UI.PXBaseDataSource.TryExecutePendingCommand(String viewName, String[] sortColumns, Boolean[] descending, Object[] searches, Object[] parameters, PXFilterRow[] filters, DataSourceSelectArguments arguments, Boolean& closeWindowRequired, Int32& adapterStartRow, Int32& adapterTotalRows) in c:\Git\Main\NetTools\PX.Web\UI\Controls\Data\DataSource.cs:line 2234
at PX.Web.UI.PXBaseDataSource.ExecuteSelect(String viewName, DataSourceSelectArguments arguments, PXDSSelectArguments pxArguments) in c:\Git\Main\NetTools\PX.Web\UI\Controls\Data\DataSource.cs:line 1762
```

What other logs that may helpful?

Windows logs

When application restarted suddenly

Installer Logs

When you have unsuccessful installation

SQL Server Profiler

SQL Related Operations, Commands

Brower Developer Tools

UI related issues

MSMQ Journal

Push Notifications and Business Events related issues

Acumatica Logs

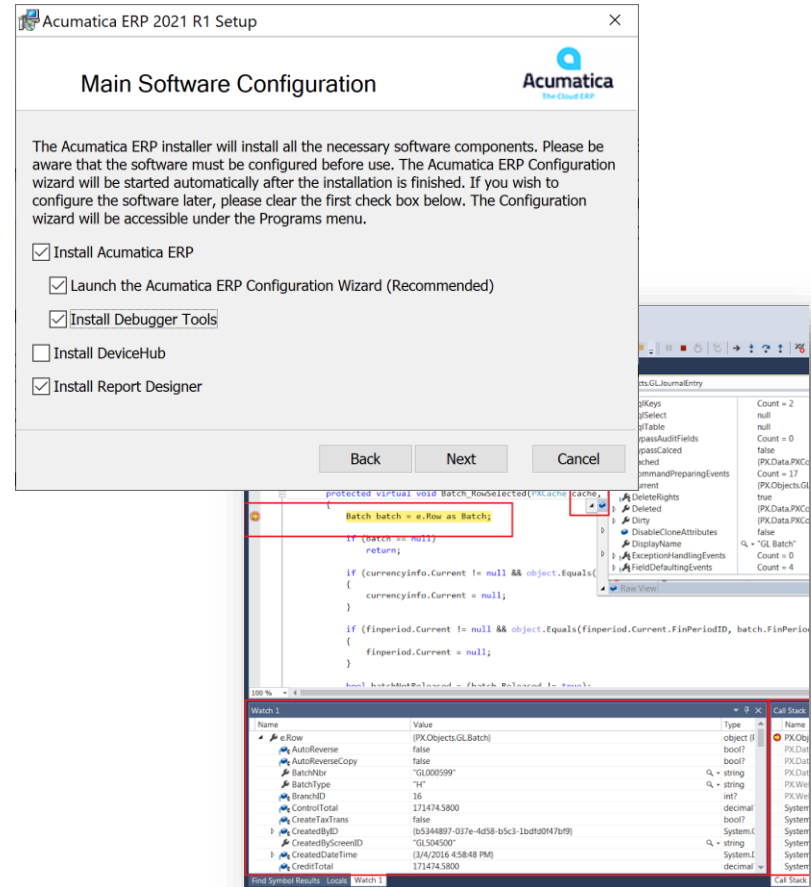
- **AUScheduleHistory** – all history of scheduler execution.
- **AUAuditHistory** – all history if fields changes if enabled.



Debugging with Visual Studio

Debug Acumatica Code

1. Install Debugger Tools (from Installer)
2. **Debug=True** in **web.config**
3. Copy **PX.Objects.pdb** to site\bin
4. Disable “**Just My Code**” in Visual Studio
5. Add Breakpoint
6. Attach Debugger
7. Debug!



Visual Studio Debugging Tips

Web.config:

- Enable Debug Web Site - `<compilation debug="True" ... />`
- Ignore Scheduler - `<add key="DisableScheduleProcessor" value="True" />`
- Optimize Start-up - `<add key="InstantiateAllCaches" value="False" />`
- Optimize Start-up - `<add key="CompilePages" value="False" />`

Visual Studio Settings

- Just My Code = Off
- Break on Exceptions + Locals / Immediate
- Attach Debugger to the W3WP

Breakpoints in DAC Setters are extremely powerful!

Process Dump Files

The image shows a Windows Task Manager window with the 'Processes' tab selected. A right-click context menu is open over the 'w3wp.exe' process, with the 'Create Dump File' option highlighted. A large blue arrow points from this option to the Visual Studio window.

The Visual Studio window is titled 'ServiceManager.cs (Debugging) - Microsoft Visual Studio (Administrator)'. It shows the 'w3wp.DMP' file loaded in the 'ServiceManager.cs' project. The code editor displays a snippet of C# code:

```
System.Threading.Thread.Sleep(Convert.ToInt32(config));
try
{
    foreach (Type table in Tables)
```

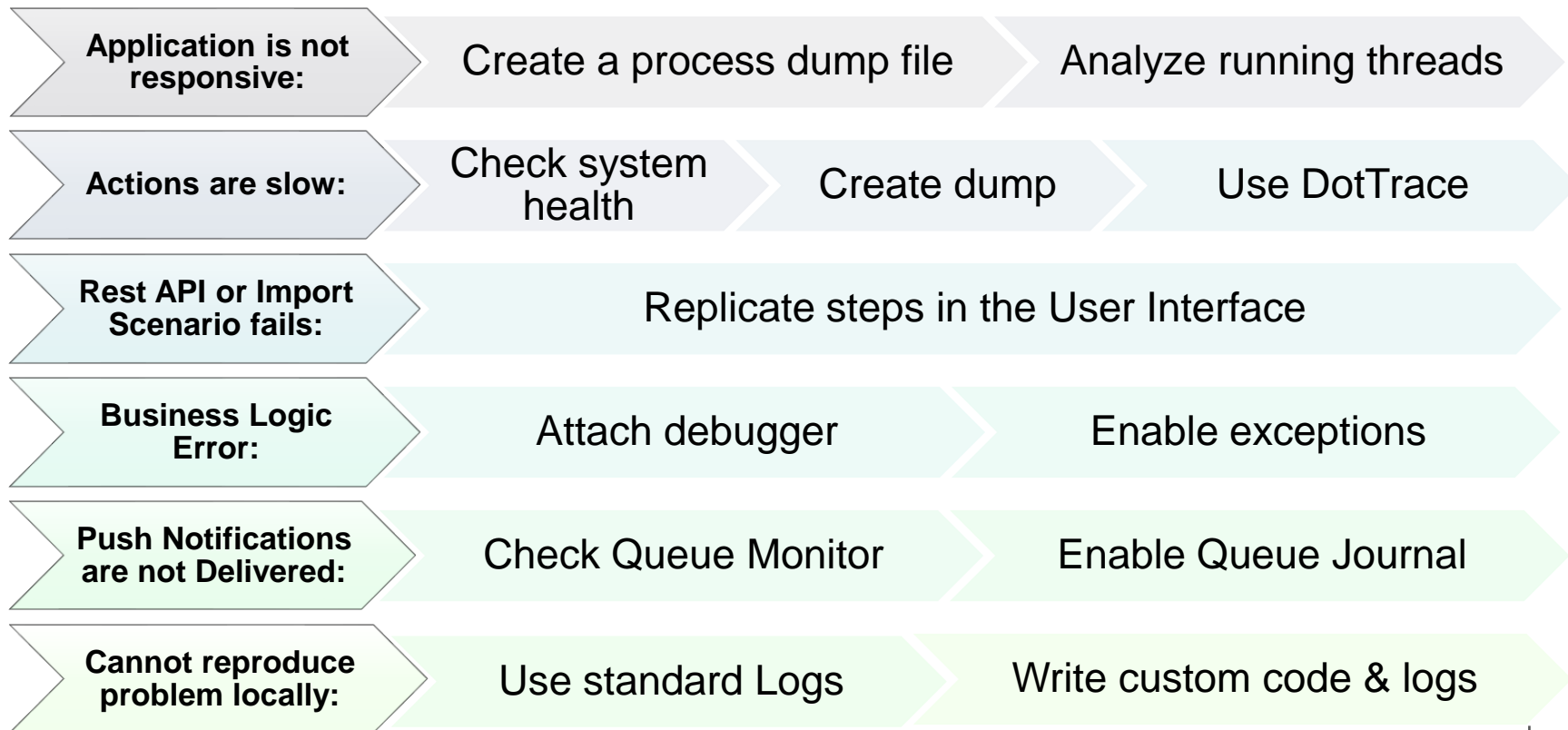
The 'Threads' window is open, showing a list of threads. The thread with ID 141236 is selected, which is a 'Worker Thread' with a managed ID of 6. The 'Call Stack' window is also open, showing the call stack for the selected thread. The call stack includes the following frames:

- ntdll.dll!00000000775515fa()
- [Frames below may be incorrect and/or missing, no symbols loaded for ntdll.d
- KERNELBASE.dll!0000007fefd301203()
- [Managed to Native Transition]
- PX.Data.dll!PX.Api.Soop.Screen.ServiceManager.InstantiateAllCaches(object obj) C#
- mscorlib.dll!System.Threading.ExecutionContext.RunInternal(System.Threading.Execu
- mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.Thread.Execu
- mscorlib.dll!System.Threading.ThreadHelper.ThreadStart(object obj) + 0x5d by
- [Native to Managed Transition]
- kernel32.dll!00000000773f59ed()
- ntdll.dll!000000007752c541()

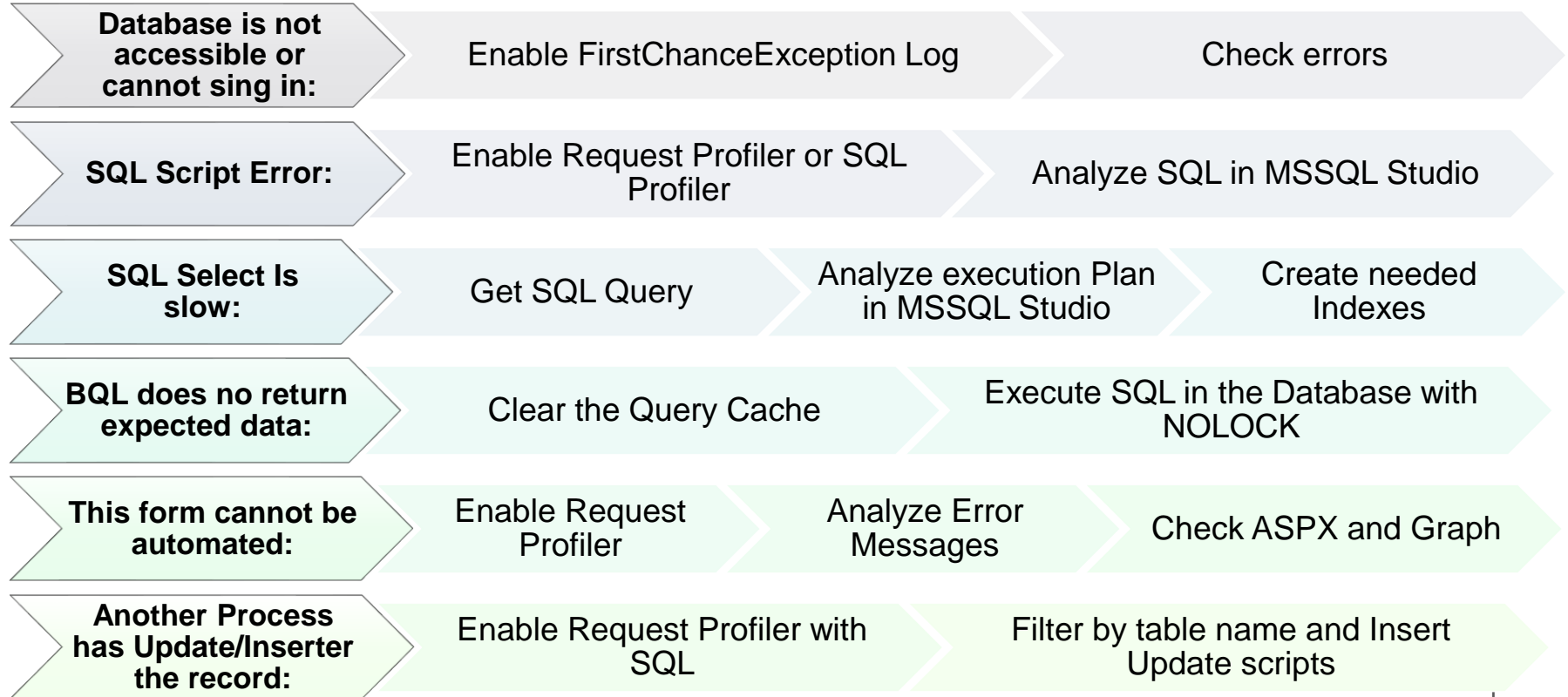


Debugging Tricks

Tricks and Best Practices



Tricks and Best Practices



Summary

Summary

- Debugging is a reverse engineering process
- Focus on unwinding of the call stack to get sequence of actions
- Search for the root cause of the issue
- Leave breadcrumbs with comments to your colleagues
- Use tools tricks!
- Debugging is a great fun!

Call to Action

- Get Involved in our Developer Community!
- Join us on **stackoverflow**: <https://stackoverflow.com/questions/tagged/acumatica/>
- Check <https://www.acumatica.com/developers/> regularly!
- Read our developer blogs: <https://www.acumatica.com/blog/category/developers/>
- Contribute to Community Projects: <https://github.com/acumatica/>
- Attend Summit 2021 | Hackathon 2021: <https://summit.acumatica.com/>
- eMail (mfranks@acumatica.com) if you have any questions or have ideas to improve



Thank You!

Sergey Marenich

Retail-Commerce Edition Team Lead / Architect

smarenich@acumatica.com

acumatica.com/developers