Investigation Scripts

Mike Shaw

23/09/2021

Contents

- **Investigation Scripts: Introduction**
  - Why
  - What
  - When
  - DEMO
  - How
  - Additional logging
  - Warnings
  - Examples
  - Where

- **Q&A**

DISCLAIMER

- These scripts are provided "as-is" and are not supported nor certified by Sage
- No warranty is expressed or implied
- Use at your own risk
- Always review the scripts before using them to ensure they are suitable for your own circumstances

Why

- Why use investigation scripts
  - Provides standard output formats for common scenarios
    - Easier to pick out the required data once you get used to the format
  - Some customers are not skilled in SQL Server/ Windows Server techniques so need detailed guidance
  - Screen sharing is not always possible in order to see issues firsthand
    - Intermittent issues
    - Secure sites
    - Timezone/language
  - Not always easy to quickly gather data on screen shares and analyse interactively
  - Sometimes a lot of data needs to be gathered for an investigation, at multiple time points, so easier to extract in bulk and analyse locally
      - Easier to package up and sent to Sage Support!

What

- What do the scripts do?
  - Execute CMD, Powershell and/or SQL scripts (Shell scripts on Linux platforms)
    - Run via Windows batch scripts (.cmd) which can be launched from Windows Explorer interactively or scheduled to run periodically
    - Deliberately kept as simple as possible
      - Don’t want to have to troubleshoot the troubleshooting scripts !
      - Security conscious customers will want to vet/understand the scripts
      - Some scripts need site specific modifications
    - Self documented, so read the scripts to understand what they are doing
    - Most scripts are READ ONLY (no modifications to target system)
      - With a very few exceptions !
    - Batch script provided to run several scripts in one go
      - Can be scheduled with Windows Task Scheduler if needed
    - Gotcha’s
      - “Run as Administrator” needed in couple of places (See script documentation)
      - Takes account of UK and US dates formats but may not fully work with other date formats… (let me know!)

When

- When to use these scripts
  - Probably most useful for these types of issues :
    - Performance
    - Intermittent
    - Where not yet focused into one likely root cause
    - Complex, e.g., multi-node installs
    - Can also just be used to gather log files and config files

- Steps to take when requesting a customer run the scripts
  - Decide which of the scripts are useful to suit your circumstances
    - No point in running any script that will not provide some useful evidence
  - Provide detailed action plan for customer to follow
    - Mention any scripts that need to be edited for site specific parameters
    - Make clear when to run (once only, regularly)
  - Suggest you send only the specific scripts to be run to the customer

DEMO

- Unzip scripts
- Run setup commands
- Check and run mzBatchRun.cmd
- Create and run schedule
- Collect data files

How?

- How do I implement the scripts

  - Make sure you are using the latest script versions
    - The scripts are always being improved, so best to regularly get the latest set

  - Fine tune the scripts you want to send to customer and then ZIP up to give to customer
    - Keep the same file structure
    - You could send all scripts to all customers, but I prefer to take out some of the lesser used ones at least
    - Consider if you need to edit the scripts to make them more specific to your needs

  - Copy the ZIP file to the server(s) from which you want to collect data
    - Where multi-node installation, you may need to install and run scripts on more than one server
    - Suggest to extract into subdirectory on a disk which has a fair amount of free space e.g. “D:\SageSupport” then extract the ZIP file into subdirectory “InvestigationScripts”
      - The scripts and their output shouldn’t take a lot of disk space, but does use temporary disk storage to package up the outputs

    - Go into “..\InvestigationScripts\setupFiles” directory
      - Run “mzCreateClassicEnvRun.cmd” and “mzCreateTechStackEnvRun.cmd” (You can also run “mzCreateLocaleRun.cmd” if you want, but don’t have to)
      - Check the created files “mzSetTechStackEnv.cmd” and “mzSetClassicEnv.cmd”
        - Make sure they look reasonable
        - Need to change database DB_NAME and SCHEMA_NAME

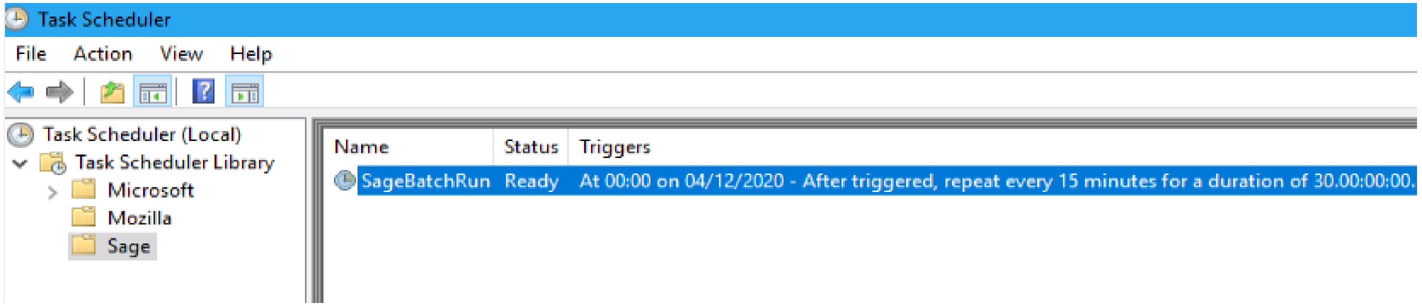
How?

- How do I run the scripts
  - Start by reading the provided “InstallationAndUserGuide.pdf” !

  - In general, there is a “.cmd” file to launch either Powershell, Windows CMD or SQL script (“.sh” for Linux)
    - These can generally be run either interactively or scheduled

  - To run several scripts together at the same time, use the “mzBatchRun.cmd” located in “..\InvestigationScripts” directory

  - To schedule the scripts using Windows Task Scheduler:
    - Check/edit the “mzScheduleTask_mzBatchRun.ps1” file
      - mzRepeatInterval and mzRepeatDuration may need adjusting
    - Run “mzScheduleTaskRun.cmd” and say “Yes” when prompted (as it needs to run “As Administrator”)
    - Launch Windows Task Scheduler and check there is a “Sage” section with task name “SageBatchRun”



Name	Status	Triggers
SageBatchRun	Ready	At 00:00 on 04/12/2020 - After triggered, repeat every 15 minutes for a duration of 30.00:00:00.

How?

- To schedule the scripts using Windows Task Scheduler (Continued):
  - Go to the tasks “Properties”, notice the user defaults to SYSTEM
    - This does not allow you to call URLs, or anything needing network access such as SQL scripts
    - Most of the time you probably should change the user to a named windows account that has “Log on as a batch job” (such as the X3 Runtime account)
  - Click the “Run” button and check the results are as expected

- How do I check the script output
  - Check the script output directories, and file contents
    - Powershell\output
    - MongoDB\MongoOutput
    - SQL\SQLoutput
    - Maybe others….

- How do I quickly gather up all these log files
  - In the “..\InvestigationScripts” directory, run the “mzGatherLogs.cmd” script
    - Check/edit the script first to gather just what you need
    - Must use the “Run as administrator” option to ensure you get everything

Warnings

- These scripts are normally tested on Version 12
  - Should work without modification on Version 11
  - Most scripts work on PU9 but there are some PU9 specific scripts included to replace those that are known not to work
- Understand what the scripts do before you tell a customer to run them
  - Run/check on your own internal test system first
  - Is useful to have some idea what data you want to collect and what might be useful, before you run the scripts

Additional logging

- Don’t forget to incorporate any other additional logging that may be useful

  Navigate to Administration, Administration, Settings, Global Settings

  In the logs section, set the following parameters to ensure the log files can grow big enough for the logging needed, and the log files do not get removed too quickly. These suggested settings assume you have the default values currently set:

  - Maximum file size (MB) set to "200" (10 default), Maximum number of files per day "5" (5 default), Maximum number of days "14" (5 default)
  - Enable web client logs (Checked) , Web client logs level = **Info**
    - In the "Server Logs" section, set the following sections to "**Info**" log level ***(Trim these options to suit your own requirements)***

    --- batch
    --- http--> in
    --- http--> out
    --- loadBalancer--> http
    --- loadBalancer--> session
    --- login--> accessRights
    --- login--> authentication
    --- login--> user
    --- orm--> mongodb
    --- session (No longer there from V12 p26)
    --- x3Comm--> client
    --- x3Comm--> conn
    --- x3Comm--> dispatcher
    --- x3Comm--> loadBalancer
    --- x3Comm--> pool
    --- x3Comm--> print
    --- x3Comm--> runtime (V12 P26+)

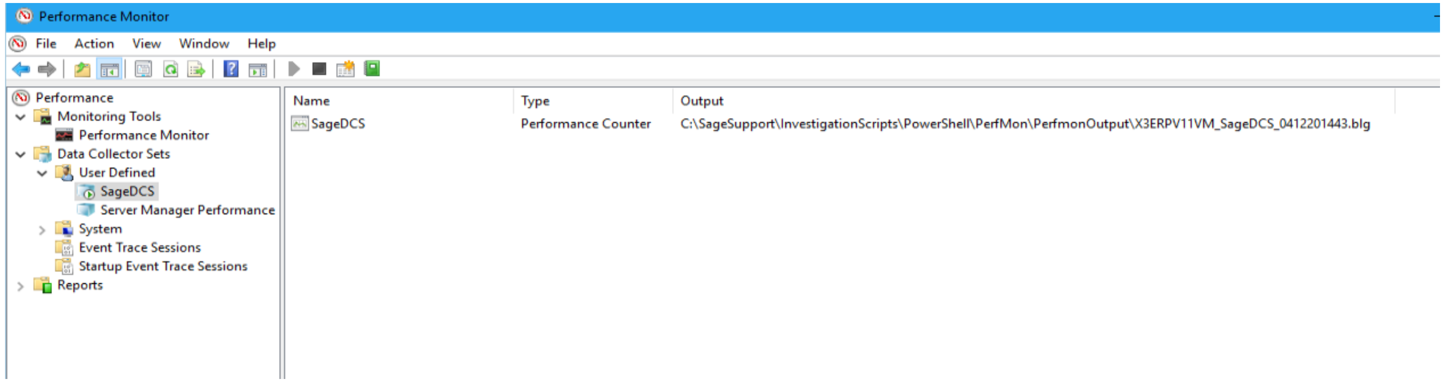
- In the "Server Logs" section, set the following sections to "**Debug**" log level

  --- console
  --- memory
  --- System--> config
  --- System--> monitor
  --- System--> process

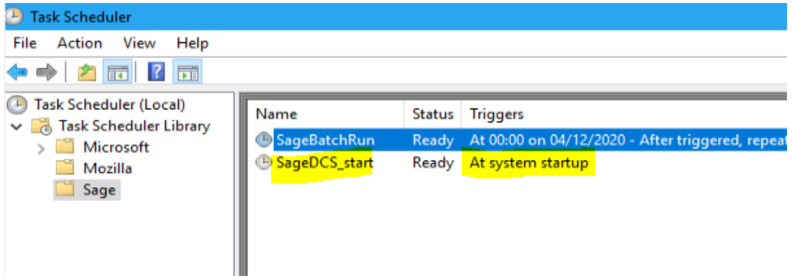
NOTE: these changes take effect immediately. Ensure you remember to reset these back to the original values, once the troubleshooting process for this issue has been completed

Example – Windows PerfMon

- Setup Windows Performance Monitor - Data Collector Scripts (DCS)
  - “..\InvestigationScripts\PowerShell\PerfMon” directory, read and implement “mzSageDCScreate_README.txt”
    - Use SQL Server Management Studio (SSMS) to check the instance name
    - Edit “mzSageDCScreate.ps1” and change “$mzINSTANCENAME” on line 27
    - Run “mzSageDCScreateRun.cmd” using "Run as administrator"
    - Launch Windows Performance Manager
      - Data Collector Sets> User Defined
      - Check the "SageDCS" script is running and note the output directory/filename
    - Check Windows Task Scheduler and notice it launches the “SageDCS” on system restart
      - Make sure this is what you want to do
    - There is no way to guarantee PerfMon data is written out when Windows Server is shutdown, so you may lose the data in the last file when the Server is shutdown



Name	Type	Output
SageDCS	Performance Counter	C:\SageSupport\InvestigationScripts\PowerShell\PerfMon\PerfmonOutput\X3ERPV11VM_SageDCS_0412201443.blg



Name	Status	Triggers
SageBatchRun	Ready	At 00:00 on 04/12/2020 - After triggered, repeat
SageDCS_start	Ready	At system startup

Example – MongoDB data

- You want to take interactive backup of MongoDB data before applying a Syracuse patch
  - Shutdown Syracuse
  - Run the script “.. \InvestigationScripts\MongoDB\mzRunMongoDump.cmd”
    - This dumps all MongoDB collections into sub-directory “mongoDump_backup”

- You want to see what data is in MongoDB “User” collection
  - Either:
    - Run the script “.. \InvestigationScripts\MongoDB\mzMongoShell.cmd” then use manual interactive MongoDB shell commands, e.g.

```
Use Syracuse
db.User.find().pretty()
```

    - OR edit then run “mzRunMongoExport.cmd”
      - Check/change “CollectionsToExport” on line 24 to include “Users”
      - This dumps ASCII file into sub-directory “mongoExport_backup_<DDMMYY>_<HHMM>”

Example – Check X3 settings and SQL data volumes

- I want to check some X3 parameters and see data volumes in some of the SQL tables
  - Check the existing SQL scripts first to see if any will give you what you need
    - By happy co-incidence, you find that “mzX3Parameters” and “mzX3TableRowCounts” will give you exactly this information
  - Edit then run the “mz_RunSQLGeneral.cmd”
    - Change the “SCRIPTNAMES” to list the scripts you want to run
  - The output goes into sub-directory “SQLgeneralOutput”

Where

Sage UK Business Partners (only) can obtain the scripts via GitHub

- Setup yourself with GitHub account, if not already got one you can use
- Email me your GitHub username (from your work email address) and I will invite you to join the “sagex3scripts” organization
- Once you are a member, you can go to the following URL and/or use Github Desktop (https://desktop.github.com) to keep in sync with any changes to your local laptop
  - https://github.com/sagex3scripts/InvestigationScripts

If you are not a UK Business Partner, please contact your local Sage support office.

Any questions, suggestions, corrections, improvements or any other comments or ideas regarding these Investigation Scripts, then please let me know by emailing me at Mike.Shaw@sage.com

Thank you

sage