FSM Performance Troubleshooting Guide

Technical guide to locate sources of performance degradation and provide known fixes

IFS

Contents

FSM Health Check Tool

For both health checks and support issues it can be very beneficial to run the FSM Health Check Tool initially. The results of the tool can then be used alongside this document to better target areas of improvement and save some time on investigations.

The Health Check Tool is an internal IFS tool used to run several diagnostic tests against the environment database to quickly determine common settings, configurations and data that can either cause or point to the cause of inefficient environments.
The tools and its associated quick start guide can be found here for IFS employees. Please read the guide before running the application as it and the tool are regularly updated to keep in touch with the latest known information.
For partners and customers who require a health check report, please request this via your IFS contact.
For assistance interpreting particular results of a health check report please create a post on Community detailing the report result you are having trouble with and the question you have regarding it.

Setting Scope

The first step in any performance check of FSM must be setting the scope and agreeing a defined, quantifiable success metric.

There are different obstacles when setting the scope and success metrics depending on whether the performance review originates from a support request or as a general health check of the system.

In both cases, though, the aim should be to determine what the business priority is in terms of performance and to give a commitment on best efforts but not to a specific speed or timing.

E.g. 'Focus will be on ensuring new tasks reach the engineers device as fast as possible' rather than 'all screens will load in under 3 seconds.'

Setting scope on support issue performance investigations

The scope and success metrics can be easier on a support issue investigation as this can be easily constrained to the area that triggered the support request and a return to the previous level of performance prior to the trigger.

It is still important to confirm this before starting and not rely on assumptions, however, as often memories of previous performance differ depending on peoples experience, priorities in the system can differ and investigations into the original issue can sometimes reveal greater underlying issues with higher impact.

Setting scope on Health Check performance investigations

Health Checks for FSM can be highly problematic if no scope and success metrics are set early. FSM's high level of flexibility trades off against a referenceable performance level.

This may require a lengthier discovery period prior to investigation to work with the customer to confirm where their absolute priority business areas are.

Often the customer will be looking for IFS to hit a particular target due to a limitation from elsewhere in their company (either a commitment they have made to their customers or another applications requirements). We should not guarantee this will be met but we can agree to use it as something to aim for.

Upgrades

Usually there is a trigger for a drop in FSM performance and often this can be tracked down to a recent upgrade.
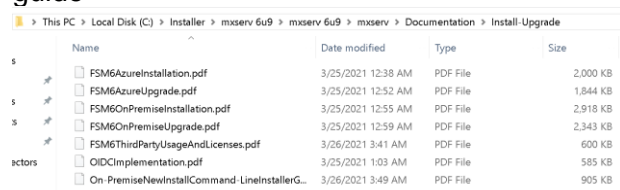
It is important to check early on in a performance investigation if there has been a recent upgrade at the time the performance dropped.
The customer should be able to advise this if asked.

There should have been a levelling of testing in a relevant test environment prior to an upgrade of the production environment but often performance issues cannot be accounted for in a test scenario.

The first step is to confirm who performed the update. If this was performed by IFS, we can check internally with the individuals who did this if there were any errors or anything out of the ordinary that may account for the performance change.

If this was performed by a partner or by the customer themselves it should be checked if the upgrade was done as per the installation guide

(copies are held in the product updates area of community.ifs.com) packaged with the installer as deviations from the guide can lead to unstable setups that do not leave much evidence short of a full, thorough investigation of the system. Be sure to use the guide that relates to the version and the type (on premise, Azure etc) of installation.
In these cases, where changes have been made, it is best to try the upgrade again following the install steps and see if this corrects the issue.

The next step is to check the release notes for the new version for any changes to areas particularly affected by the performance drop. The release notes are again held in the product updates area of community.ifs.com.

If the upgrade has been performed as per the documented instructions and there are no relevant details in the release notes, then the originating area for the performance drop needs to be discovered (if not already known)
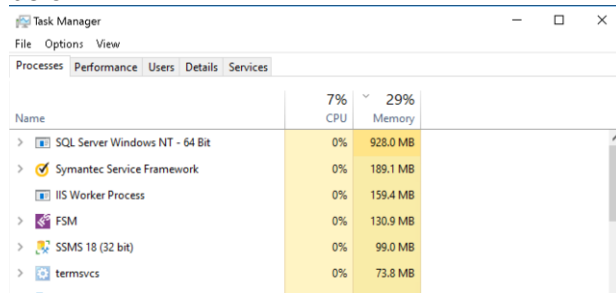
Finding the Origin

Before FSM is interrogated fully it is best to confirm the main origin of the performance issue. It may be that there is some other process or application, separate from FSM's workings but within the same infrastructure, causing enough issues on the server to knock to the FSM performance as well.

Find the Problem Process

To confirm the issue is within the FSM process itself the processes on the application server need to be checked.
Once on the server open the task manager by right clicking the windows tool bar and selecting task manager, search for it in the search at the bottom of the screen or select task manager from the server lock screen.

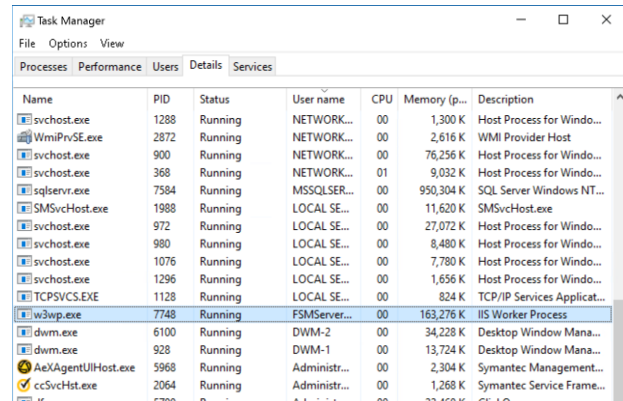Once open you should see something like below:

If the issue is on this server then you would expect to see either CPU and/or Memory to have near 100% above them.

Click the column that is near 100% to order the processes by how much CPU or Memory they are using. The top of the list is likely where the issue is coming from. The FSM process is called FSM and has the IFS logo so it should be easy to spot (can be seen in the screenshot above).

An alternative or good additional step is to view the details tab on the task manager window:

It may be that the main FSM process is not the top running one, but it could still originate from FSM if one of the IIS Worker Processes is high.

On the details tab you can see these are called w3wp.exe and in the Username field it will indicate that it is for FSM as seen in the screenshot above.
You can, again, order this screen by CPU or Memory to make it easier to find the problem process.

If the app server is showing usage below the 80% area for CPU and Memory, then it may be that the issue is coming from another server. The above steps should then be repeated for each app server and then, if there is still no issue found, on any database servers. If the database server shows the Oracle or SQL database as the top process, then a trace will need to be done to determine the origin as there could be multiple databases for multiple different applications in the same setup.
Tracing is covered in the database section of this guide later.

What if the Problem Process is Not Related to FSM?

If the top process on the maxed-out server is not related to FSM, then this will first need to be worked through with the relevant support for this third-party app.
As this is separate from FSM there is not much IFS can do other than provide the detail of the process to the customer and why it has been indicated as a source of concern for the customer to then investigate with any relevant external support teams.

If IFS support the server environment, then the investigation should be handed over to the relevant team although that falls outside the scope of this guide.

If the performance is still an issue after the problem process is resolved, then investigation can restart but the scope and success metrics should be reviewed and confirmed to still be relevant.

What if There is no Problem Process?

It is possible to get impaired performance without this showing as an intense use of CPU or Memory on any server.
Often these issues will be setup related such as a firewall blocking traffic, a service not running or incompatible versions or types of OS, drivers, databases etc.
In these cases, continue through the guide but with a special attention to comparison to baseline setups. The requirements for servers and databases can be found in the installation guide referenced in the Upgrades section.

Initial Server Checks

There are some quick steps that can be taken at the point that the process is identified as problematic. These may not solve the issue but will provide helpful information to guide the investigation down the line.

Retrieving a Dump File

If a problem process is identified, then the first step should be to attempt to retrieve a dump file of the process whilst the performance issue is occurring. This will give information on exactly what is going on in the process.
To do this right click on the process in Task Manager and select Create dump file
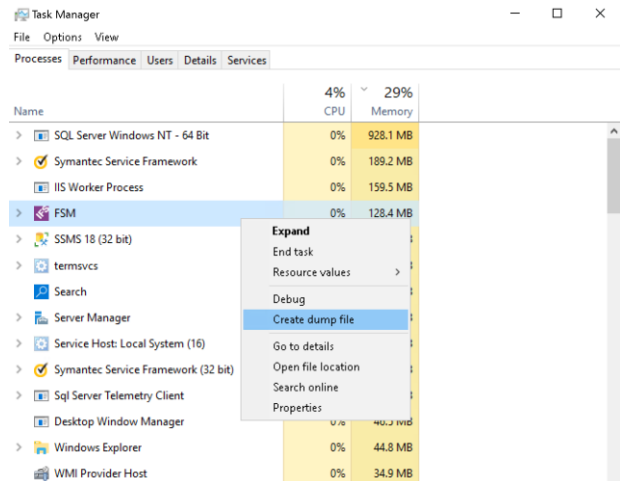
This can take some time as the server will already be slow due to the performance issue. If it cannot be obtained, then the other logs will need to be used to fill in for the dump file.

Internet Information Service (IIS), App Pools and Recycling the Pool

The next step is to find the app pool for FSM, check it is running and recycle it to see if this relieves the issue if only temporarily.
You should be able to find IIS in the below location in the server structure:
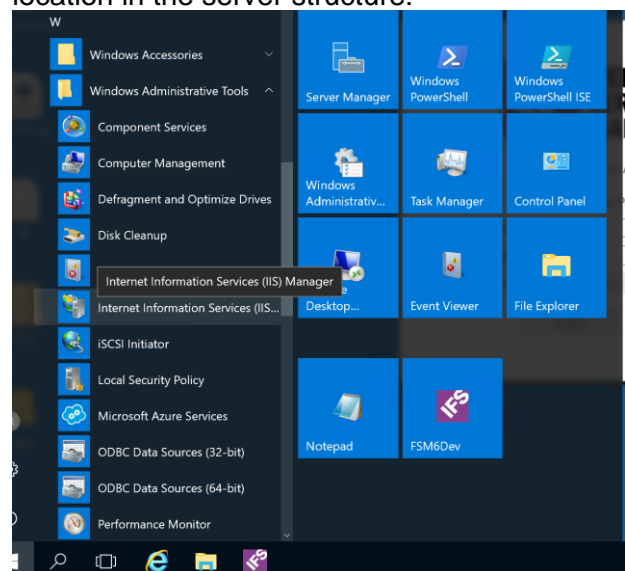
Within here can be found the application pools that FSM uses to process its transactions:
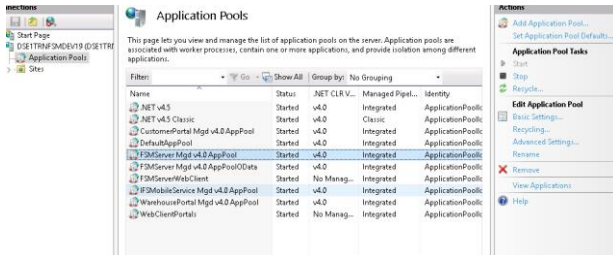
There should be one for the main FSM service and one for the Mobile service.

Check that these are both showing a status of started. If they are not, highlight the line for the FSM pool that is not started and click start from the options on the right-hand column. Check that this starts and remains running.

The app pools can be stopped and not restart correctly when there is an issue on the server that cause a restart, so it is best to check the windows server logs if the pools are found to be stopped. This is covered later in the guide.

If the pools are started, highlight the FSM app pool, and then click the recycle button on the right. Do this for each FSM pool. You should see the server performance improve within Task Manager shortly after this but keep monitoring this for a while following the recycle.

## Non-Constant Performance Issues, Server Schedules, and Logs

Sometimes the performance will spike straight back up to where it was once the pool has full started up again and sometimes the performance will stay at a reasonable level. If the performance does stay improved, then this points to an issue that was triggered by some specific action rather than an underlying issue of the setup or data. These could be scheduled processes, data imports or a single user action. The system should be monitored for a reoccurrence whilst the time and date of the original performance drop is compared against the logs and schedules to try to determine a possible instigating action.

If the performance spikes again after being improved for at least a few hours, then the date and time of the next spike should be

noted and compared in the logs and schedules to see if a similarity can be found for the two times.

The server level schedules can be checked by opening the task scheduler. FSM schedules tend to be setup within FSM (this is reviewed later in the guide), but it may be that something has been set up at the server level, so it is worth a check.

The Task Scheduler can be found by searching the server applications and will look as below:
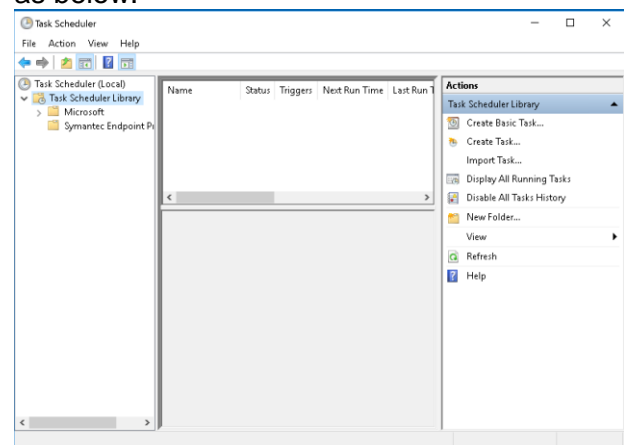
The important field will be the last run time. Compare this for any schedules against the start of the performance issues. Make a note of the next run time and monitor the system at that time to see if the performance issue reoccurs to confirm the schedule as an issue.

The Event Viewer holds all the important issues that occur on the server and can again be found by searching the applications on the server for event viewer.

It will look as below:
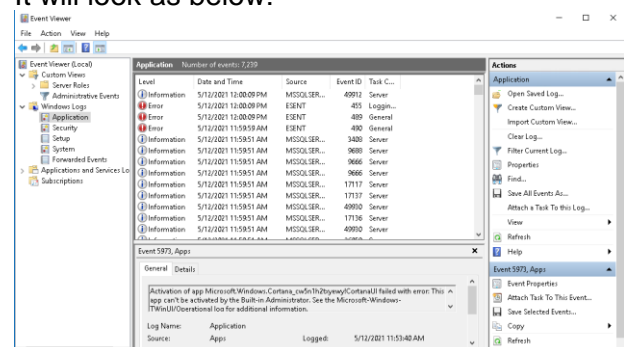
Attention should be given to items at the level of Error and occurring around the time the performance dropped.

If an error or schedule is identified but assistance is required determining the cause then feel free to contact the ACH for assistance on understanding the messaging and identifying next steps.

Database Tracing

The database can also be traced directly to try to find the particular transaction that triggers the issue. Database trace results will often contain recognisable FSM tables and fields that can be used to track down the area for further investigation.

Starting a Database Trace

Open the Microsoft SQL Server Management Studio and from the Tools menu options select the SQL Server Profiler.
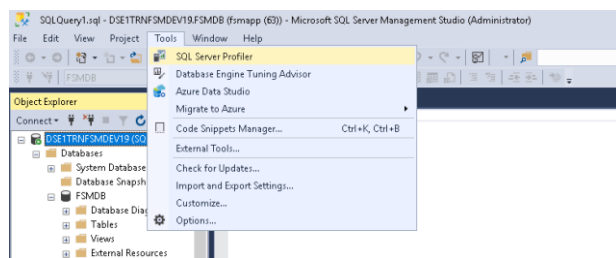

In the settings screen for the trace you can either leave the settings as they are or try the below settings:
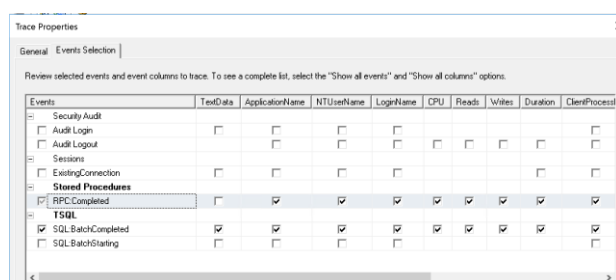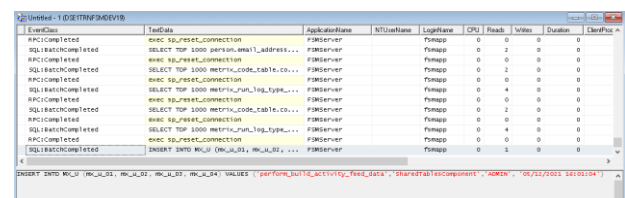

This will ensure that only completed scripts are returned and will allow you to sort the results by the duration of the transactions.

If there are too many transactions coming through on the trace you can also add a filter on duration to ensure it has a value of greater than 1000.
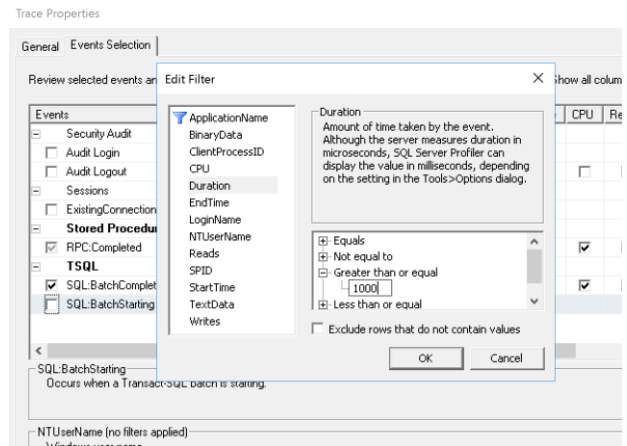
This will only find transactions taking more than a second.

Once the run is started you should see a screen like below:

Highlighting a trace line in the top half of the window will show the full script in the bottom half. It is best to run the trace for a little while, then stop the trace and review the transactions once the trace is stopped and no longer adding lines.
If no transactions are showing with a value in the duration column vastly higher than the other transactions, you may need to recycle the app pool whilst the trace is running. This will stop any long running transactions and make them appear in the trace. This should only be done if the run logs have been checked in FSM and it has been confirmed that any ongoing processes are not progressing by monitoring for a while.
Find any transactions with a greater duration than 1000 and record the script associated for review.
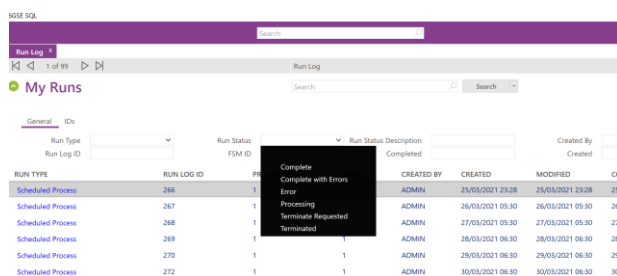
FSM Schedules and Logs

As well as the server level information, FSM holds a lot of records on what the system has been done and when. These need to be checked alongside the server to get a complete picture of the situation.

FSM Run Logs

A first port of call in the FSM app when there is a performance issue would be to open the FSM run logs. This can be found easily by searching 'run logs' on the FSM menu search.
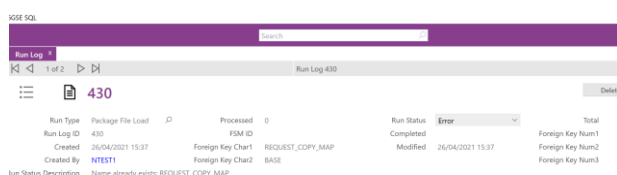
The runs that will be most interesting will be the ones with a status of error, completed with errors and processing. The results can be filtered for these statuses on the search screen.

If the performance is dropping at a similar time each day or week etc then you can also search by the created field or order the search by created and then look for the time when the performance dropped.

Clicking in a particular run log will give a screen with some more detail. On logs with errors, a text detail of the error will be displayed in the run status description area which should assist in determining the cause.

Otherwise the key to look for is a reoccurrence of the same type of run failing to complete or generating errors or always running at the time the performance drops. When that is identified you can then check created by on the run log and check with that user as to what specifically was being done on the run.

Run types of particular interest would be data loads or generation runs as these tend to be greater in size and scope. The scheduled processes are also logged here so, if the issue is reoccurring at a regular interval the scheduled process logs would be a more likely starting point.
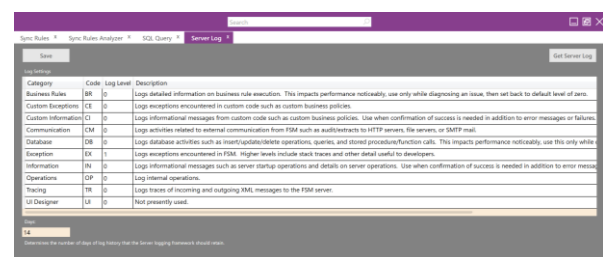
Troubleshooting for Problem Runs

If a particular run type is suspected, then a good troubleshooting step to help confirm this is to separate the full run in to two halves or even quarters. If this does not cause as bad a performance dip, then the aim should be to find ways to restrict these runs from being too large.

This could be by running the particular process more often if it is a run that processes all records since the last run or to create multiple transactions to run sequentially based off of some separation point in the data e.g. invoices with customers a-m and then customers n-z in a separate run the next day.

Server Logs

Within FSM you can also search for the Server Logs

Much care must be taken when setting these, but they can help to compliment the other logs and records to determine a cause.
The most useful and least detrimental would be to set the tracing log to 6. This will not give details of exactly what is going on in each transaction, but it can show where the same transaction is repeating endlessly in the log and when transactions do not complete.
Business Rules is another one that can help as tracking down Business Rule issues can be tricky where many are set up in the system.
This should be used with care as it will impact performance and ideally only once the general area of the system causing the issue is found to pinpoint the specific rule.
The log should be set at 9, the area under investigation triggered and then the log set back down to 0 straight away.
Save the settings after each change for them to hold.
Once the trace is done you can click 'Get Server Log' to retrieve the latest log for review.

If the logs have been on for a while (only advised for tracing type) then the historical logs can be found at the below location:

Known Areas for Easy Wins

There are a few areas of the system that can be quite CPU intensive but are little used. It can be that users are not even aware a piece of functionality exists, but it is still stopping the system. This section details the worst offenders of these.
In health check cases this is very good first port of call for some quick wins, but they have also been known to be the primary cause of support issues as well.

Team Task Syncing

By default, FSM will sync tasks to mobile devices where there is no individual assignee but there is an assigned Team.
It will sync a task set in this way to all devices for all members of the associated Team.
For environments where PSO is being used to assign tasks and compute optimum work, there is only ever one assignee on a task or where technicians are not expected to select their own work it is strongly advised that this default is amended to better suit the business needs of the customer.
Full guidance on what to change and how can be found here.

Refresh Rates

Many areas of the system have

refresh rates set by app params that then trigger data refreshes in the background that the user is often not fully aware of.
These can be located by searching 'refresh' and 'interval' in the app param screen.

The results may vary depending on the FSM version.
Some will allow a time entry, and some are a straight on or off. The on and off is easy to test as you can simply switch the app param to 'N', refresh the cache, end the FSM sessions, and then restart to see if there is a performance improvement.
For time entry it would be advised to set this to once an hour or some other large value to ensure it is not triggering.
Particular attention should be given to the Scheduleboard, GPS and Activity Feed settings as well as any setting that has been changed for the default to a quicker value.

Activity Feed

The Activity Feed is not commonly used in practice as users tend to have reports or direct notification messages set up for information, they are interested in. The Activity Feed is turned on by default though so it will be trying to record and display the relevant actions to allow relevant users each time a change is made in the system. This can build up quite fast, especially on a system with a high number of users as the performance impact become exponential with each user added.
The recommendation would be for the Activity Feed to be turned off in systems experiencing performance issues. It can then be turned back on if the functionality is truly needed but not before alternatives (such as reporting or notification messages) are considered and the setup is restricted down to the minimum number of users and following the strictest set of changes. The refresh should also be set as mentioned previously at a higher level if it is turned on to offset the performance impact.

Activity Feed is not intended for all users to track all changes on a table but for a small group of supervisors to track a small, curated list of events.

Sync Rules

To help determine if it may be the sync rules causing a performance impact the last run value on the sync rules can be checked to see if this matches with the times of the performance dip. The mm_message_out table should also be checked to ensure there is not one table other than Task that has a vast number of records in the sync queue which would indicate an issue with a large data import to a syncing table. The size of the individual messages should also be checked as overly large individual transactions can point to an ill designed sync rule trying to send to much information for one table.

There are some standard scripts here that will assist with visualising these issues. It is best to run these in the database management program directly and is best to be reviewed by a DataBase Analyst (DBA). A result in the millions for the individual transaction size script would be a cause for concern and any table with more entries than the Task table could also be an issue.

There are a couple causes that could trigger a sync rule performance issue. First would be a data import into a table that then syncs the table out and would be shown as a large quantity of records in every mobile users' queue for one specific table. The creation date on the queue lines would likely match a recent data import. As a rule, a mass import should not be done on a table that is real time sync as the system will immediately attempt to send the new information to all mobile devices at the same time as processing the import causing the load to multiply by the number of active devices.
Any table that is likely to have a mass import should be set to batch_delta and the parameters of the sync should be set such that only records in the table relevant to the work currently being done by the technician is sent to the device.

Another potential cause is a join in a sync rule that was believed to be more restrictive than it is e.g. a link to the parts for a task that attempts to include tasks only at request level but then also selects all parts on all tasks related to that request. These will show as much larger individual transactions. A good step to confirm would be to revert to the FSM standard rule if it is different to see if this improves the performance and message size. After which the sync rule script would need to be redone to only select the data needed and no more than that. Extra filters usually correct this.

The next steps should only be performed by an experienced DataBase Analyst with good knowledge of the FSM data tables.

In each of these cases, the problem records will likely need to be removed from the message queue to allow the proper syncing to resume. This should be done during down time on the database to avoid errors. This should only be done after the relevant fixes are in place to prevent an instant reoccurrence of the issue.
Only remove the queued records from the table shown to be the issue. Once these are removed, the device should be allowed to sync the remainder of their queues. This will ensure no responses on work in progress are lost from the mobile to the database.
When a user's mobile device shows a clear queue for sync on both the device and the database then the user can re-initialise to ensure all relevant data is on the device. These initialisations should be staggered to avoid excessive load and only affected users need do this.
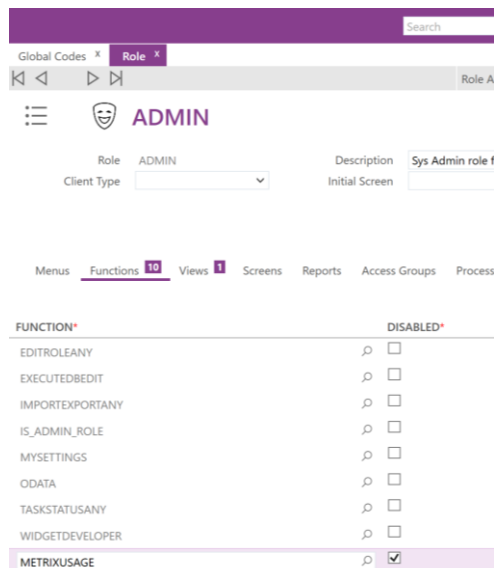
Audits and Extracts

It is worth checking if any audits and/or extracts have been setup and if these are active. If there are then confirm if there are constraints in place or they are looking at tables with low activity. An unconstrained audit/extract on a high Activity Feed can cause an impact similar to the activity feed where all work is doubled with the system trying to record all the detail of every action whilst also actioning it.

There are also alternatives such as using notification messages or reporting just as with Activity Feed to avoid the area all together.

MetrixUsage Function

Check the roles in the system for the METRIXUSAGE function

If this is found on any user role remove it. This function is designed for background tasks and adding it to any user role will steadily increase the database size until it impacts performance. Some of the system data tables may need to be cleared following this so a DataBase Analyst should be engaged to review large tables and archive. This should only be done by an experienced DBA.

Global Codes

Global codes are loaded into memory so, if they grow too large, they can cause a slow down of the system. Usually global codes would have below 100 entries so any in the multiple hundreds and up should be checked. Special attention should be paid to those codes that have hierarchies attached, especially if the hierarchy is to another large table. If a code is found with this type of hierarchy then start by disconnecting the hierarchy and see if this improves performance. After that review the codes to check if all are required or if they need to be global code.

The other code tables are not loaded into memory, only global codes, so they should not need to be checked.

Configuration Changes

Performance usually drops after some type of change. If all configuration changes are being monitored and well recorded this should just be an issue of looking up what changes were made at the time and reverting to confirm improvement. In situations where no such record is being kept, more investigation is required.

Lobbies

Lobbies can be very useful for reporting but depending on how they have been setup they can be very resource intensive. Identify any lobbies that are non-standard. The quickest way to confirm them as a source of the issue is to temporarily disable them and see if performance improves. If they are the issue, then their setup will need to be re-investigated to determine if a less frequent update or a more restricted dataset can achieve a suitable result.

Business Rules

Business Rules can sometimes cause looping when not setup ideally which will impact performance. This should be characterised by the performance dropping when loading specific screens rather than starting as soon as the application is started, at specific times or seemingly random occurrences.
The screen that triggers the performance drop will help direct to the business rule at fault but even without that there are some ways of tracking the Business Rule down.
The first would be to run a Business Rule trace as described in the of this guide.
The other would be to manually deactivate all rules and then, if performance improves, reactivate one by one until the drop is seen.
When the Business Rule is located, check that the output of the rule causes the Business Rule trigger to no longer be valid e.g. if the rule

triggers off of a field value ensure that the rule sets this to a different value on completion.
If the rule output is to run XML script then this will need to be reviewed, ideally change this to be a standard output rule and create multiple linked rules if required rather than running the script if the issue with the script cannot be identified.

Database Indexing and Structure

Sometimes the issue can come down to database fragmentation over time or database setup and configuration not being updated to match changes in the data use.

Fragmentation

> **This next step must be performed by an experienced DataBase Analyst**

There should be a regular reindexing of the FSM database scheduled to run during downtime. Check that one is setup and active. If one is not, then agree and arrange one to run on some regular schedule.
The IFS guidance is that any table with over 75% fragmented should be reindexed.

New Indexes

In some cases, it may be beneficial to introduce new indexes to account for changes in use of some of the fields in FSM. The main example of this would be user_def fields used in join or query constraints as user_def fields are not indexed as standard but if heavily used, creating an index could offset some of the resource use. This needs to be a case by case decision made by an experienced DataBase Analyst though and not a decision to be made lightly.

Large Tables

Check the database tables to ensure none are growing beyond an acceptable size. Unfortunately, there is no cut and dry size limits for the tables as they will vary from configuration to configuration and with the number of users. A comparison can be made to other working environments of similar design and an experienced DataBase Analyst should be able to determine tables that are growing too quickly or seem to large.
Once the table is corrected there should still be a check of the processes that lead to the specific table being written to so that a reoccurrence is avoided.

Mobile Performance

On some occasions the performance can be found to only impact the mobile users with the back-office staff still being able to use the application with little issue. This could be a single, group or all the mobile users.

The first step should always be to confirm which users are affected and which are not.

Mobile Dashboard

With a list of the affected users the first stop should be the Mobile Dashboard.
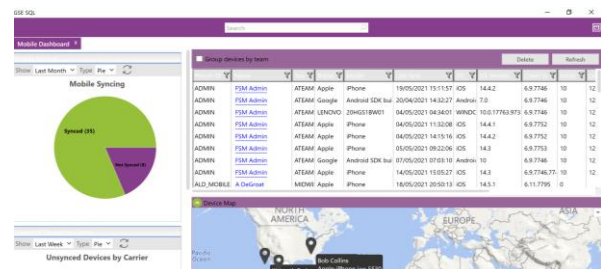
This shows a lot of the same information as the mobile tab on person with the addition of being able to see all users at once and the errors for each device.

Particular attention should be on the below
- The client versions should all be the same and match the current FSM version in use
- The OS version should be one supported by the FSM version in use. This can be referred to using the release notes in the community
- Each user should ideally only have one active device each
- Check the errors against the devices for the users affected

- Check to see if all affected users have the same device if different device types are in use
- Are the affected users situated in rural locations where networks may be slower?

Mobile Queue Checks

Use the script for analysing user queues from this article. If the top users in the results have a much larger queue than the remaining users check the last sync time. If this shows a few days back the users should be urged to sync or, if the device is no longer in use, it should be removed as a device which will also remove the queue.
If the sync time is recent then consider using push notifications and background syncing to keep the queue reducing (this is available from FSM6U5 onwards). Please see the Mobile Messaging guide for assistance in the setup and use of this functionality.
If the queue is not reducing at all despite the sync attempts then see the advice in the sync rules section.

Further Assistance & Feedback

For further assistance:
Customers – please log a new support case via the customer portal referencing that this guide has been used
Partners & Internal – please contact ACH