IBM® Security Identity Governance and Intelligence

Version 5.2.4

Performance Tuning Guide

Note: Before using this information and the product it supports, read the information in "Notices".

1st Edition notice

Note: This edition applies to Version 5.2.4 of IBM Security Identity Governance and Intelligence and to all subsequent fixpacks and modifications until otherwise indicated in new editions.

Table of Contents

About this publication

The *IBM® Security Identity Governance and Intelligence Performance Tuning Guide* provides information on tuning middleware for IBM Security Identity Governance and Intelligence Version 5.2.4.

Access to publications and terminology

This section provides:

- A list of publications in the IBM Security Identity Governance and Intelligence library.
- Links to "Online publications."
- A link to the "IBM Terminology website."

Online publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security Identity Governance and Intelligence Information Center

The http://www-01.ibm.com/support/knowledgecenter/SSGHJR/welcome site displays the Knowledge Center welcome page for this product.

IBM Knowledge Center

The http://www-01.ibm.com/support/knowledgecenter site displays an alphabetical list of, and general information about, all IBM products. Use the Search dialogue box to navigate to *IBM Security Systems* for a list of IBM Security Products.

IBM Publications Center

The http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss site offers

customized search functions to help you find all the IBM publications you need.

Support information

IBM Support Portal provides assistance with code-related problems and routine, short duration installation or usage questions. The IBM Support Portal can be found at https://www.ibm.com/support/entry/portal/support.

1. Statistics Enablement for the Database

Tracking the behavior of the database can greatly improve the ability to tune this tier for optimal performance. The following DB2 commands invoked on the database environment enable the monitor switches. The database instance must be restarted to make these effective for new connections. Additionally, these monitors should be periodically restarted to ensure the statistics do not become stale. As a best practice, they should be refreshed when planned maintenance for the product includes a database instance restart.

- db2 update database manager configuration using DFT_MON_STMT ON
- db2 update database manager configuration using DFT_MON_BUFPOOL ON
- db2 update database manager configuration using DFT_MON_LOCK ON
- db2 update database manager configuration using DFT_MON_SORT ON
- db2 update database manager configuration using DFT_MON_TIMESTAMP ON
- db2 update database manager configuration using DFT_MON_UOW ON

The switches above will report on the SQL statement, bufferpool(s), lock(s), sort behavior, timestamp information, and unit of work (UOW). Statistics related to these items will now be displayed in subsequent snapshots or output from *db2top*. The following command will report whether data collection is enabled for these statistics.

```
$> db2 get dbm cfg | grep DFT_MON

   Buffer pool                          (DFT_MON_BUFPOOL) = ON
   Lock                                    (DFT_MON_LOCK) = ON
   Sort                                    (DFT_MON_SORT) = ON
   Statement                               (DFT_MON_STMT) = ON
   Table                                  (DFT_MON_TABLE) = OFF
   Timestamp                         (DFT_MON_TIMESTAMP) = ON
   Unit of work                            (DFT_MON_UOW) = ON
```

In laboratory testing, table monitoring has been known to cause a slight performance impact when enabled. Enabling this statistic should be reserved for times when the statistic is collected, and then disabled afterwards. As stated before, the database instance will require a restart after this setting is changed.

In the PostgreSQL implementation, the statistics collection engine is controlled by the settings found in the file postgresql.conf. For the Identity Governance system, the following are set in the

standard configuration.

track_activities	on	Collects information about executing commands.	
track_activity_query_size	1024	Sets the size for pg_stat_activity.query, in bytes.	100 - 102400
track_commit_timestamp	off	Collects transaction commit time.	
track_counts	on	Collects statistics on database activity.	
track_functions	none	Collects function-level statistics on database activity.	
track_io_timing	off	Collects timing statistics for database I/O activity	

With these settings, the pg_stat and pg_statio views can be used to collect information at runtime.

To view all the settings, the DB administrator can use the Identity Governance virtual appliance (VA) command line interface (CLI). From the root menu of the CLI, the administrator will navigate to igi → postgres → postgres_cmd. The user will then be prompted to login. The following is a useful command to extract the current DB settings.

```
select name, setting, unit, short_desc, min_val, max_val from pg_settings;
```

There are several methods which can be used to access the PostgreSQL database remotely. The tool pgadmin (https://www.pgadmin.org) is a popular method, as well as the more common *psql* command. Here is an example of a remote access to the Identity Governance PostgreSQL DB from a Linux system using the *psql* command.

```
psql -h hostname -o /tmp/pgoutput.txt igidb postgres
```

The user will login with the VA admin password. In this case, output from any query run during the *psql* session will be written to a local file on the Linux system in /tmp/pgoutput.txt. Refer to the man page for *psql* for formatting techniques and methods to invoke commands for scripting.

The following are useful views to collect statistics for runtime behavior.

- pg_locks
- pg_stats
- pg_stat_activity
- pg_stat_database
- pg_stat_user_tables
- pg_stat_user_indexes

- pg_statio_user_tables
- pg_statio_user_indexes
- pg_stat_replication (For cluster environments)

To ensure the statistics are updated, it is advisable that tools such as “runstats” for DB2 or “analyze” for PostgreSQL be run periodically on long running DBs, or when the data set changes dramatically or frequently. After an upgrade from one firmware level to the next, the DB administrator should also follow best practices by running a *reorg* on the database tables and indexes, as well as the *runstats*.

In addition to the tuning provided by the *runstats* command, the DB2 administrator should also perform a *db2rbind*. This procedure should be performed each time the database changes dramatically (large bulk loads, many new applications, many new permissions or endpoints, for example), and at regular DB maintenance windows. It generally takes a few minutes to perform and should be executed on the running database. To avoid conflicts and errors, the VA application server should be stopped. The command for invoking this procedure is listed below, where -l (lower case 'L') specifies the log file.

db2rbind *db_name* -l *log_file_name* all

The table and index statistics are generated during table initialization, and for dynamic SQL, the *runstats* command will update the statistics for improved query plans. For static SQL code, statistics are not updated by *runstats*, and will become stale. The *db2rbind* command will update the statistics for static SQL, improving query plan performance. This is particularly important when the database approaches enterprise levels. Laboratory tests have demonstrated significant performance improvements using this procedure on enterprise level DB2 deployments.

See DB2 Documentation in the IBM Knowledge Center for additional information regarding the db2rbind command.

In general, DB tuning and administration best practices should be used to maximize the data tier performance. Those recommendations will be specific to the customer’s data contents, size, and usage patterns. Hints for applying indices or setting individual tunable values can be found in DB statistical data or by studying the DB log. As an example, the DB2 tunable “Maxappls” can be tuned according to the number of application connection requests. This tuning recommendation is apparent by viewing the db2diag.log.

2. Tuning the Rule Engine Scan Rate

The Rule Engine is the means by which items in the event queue(s) are processed and changes are made to the Identity Governance system. The Rule Engine behavior is configurable in the Task Planner Module. By default, the Rule Engine will scan the target event queue every 20 seconds looking for work. Upon completing a work item from the queue, the Rule Engine will return to the queue to take the next work item. If the queue is empty, the Rule Engine will wait 20 seconds to scan for newly arriving work. Under the *Scheduling* tab, the frequency of the scans can be adjusted. Note that adjusting the frequency will require the task to be stopped, then started again after the change.

The default scan rate of 20 seconds makes the Identity Governance highly reactive to changes in the system.

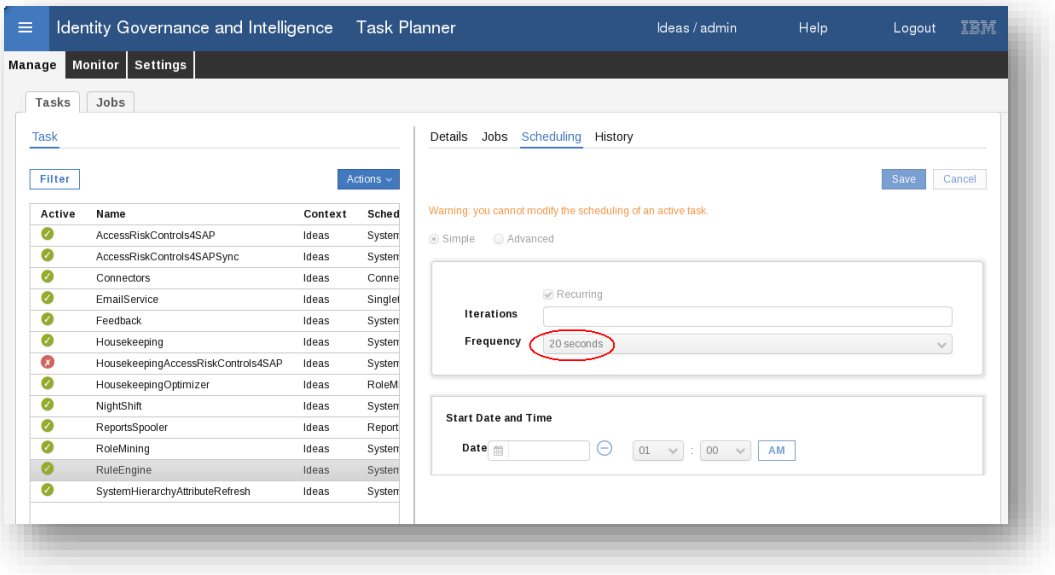


Figure 1: Rule Engine Scan Interval

3. Tuning the Rule Engine Cache

The Rule Engine will be invoked any time an event exists in the event queue. In the Task Planner Module, the Rule Engine **cacheTime** value can be used to enable the rules cache for a particular Job. The default setting for the rules cache is disabled. In disabled mode, the Rule Engine will

read the rules from the database and compile them for each event processed.

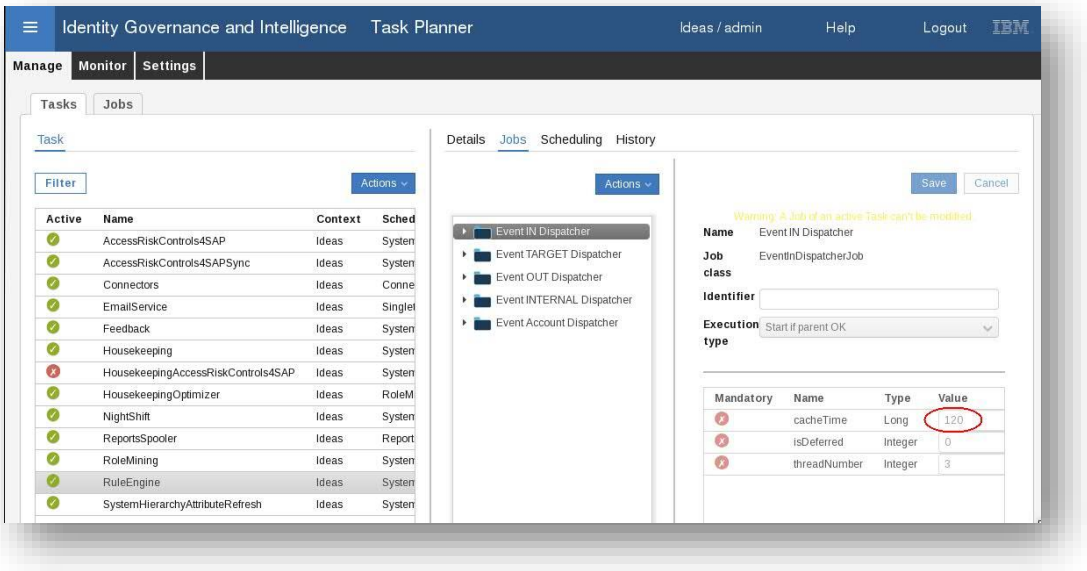


Figure 2: Configurable Rule Engine Properties

In a dynamic environment where the rules are modified frequently, a disabled rules cache ensures that the latest rules are always applied during event processing. In a stable, long running environment where the rules are established and remain unchanged for extended periods of time, caching can and should be used. Changing the **cacheTime** value to 600 minutes would result in a 10 hour cache of the current set of rules. A value of 120 minutes is used by default when the rules cache is enabled. The **cacheTime** value can be changed independently for each Job under the RuleEngine.

Using the rules cache results in fewer reads from the database, and fewer compiles of the rules, reducing CPU consumption on the VA and the database tier. Changing this parameter requires that the Rule Engine task be stopped, then started again after the change is made.

4. Task Planner

The Task Planner is the internal scheduler for the Identity Governance product. This module is responsible for all batch and background jobs. The Task Planner has six schedulers.

- System Scheduler
- Reports Scheduler
- Connectors Scheduler
- Singleton Scheduler
- Custom Tasks Scheduler

- Role Mining Scheduler

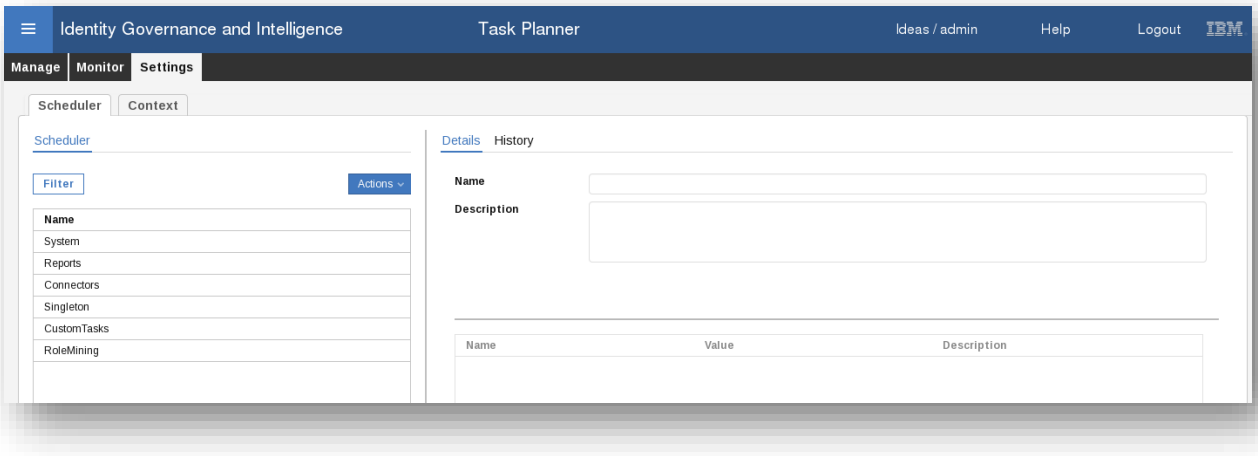


Figure 3: Task Planner Schedulers

The default number of threads for each scheduler is 2, which is also the maximum number of threads. The only exception is the Singleton scheduler which, as the name implies, has only 1 thread. Each thread can execute one task at a time. As an example, the Role Mining Scheduler can run up to 2 tasks at the same time. The number of threads per scheduler is not configurable, but the schedule for when the scheduler activity runs is configurable.

Each Task found in the Task Planner is associated with a specific scheduler which can be seen in the fourth column of the left frame. Referring to the *Scheduling* tab in the right frame will provide information about the frequency for the given Task.

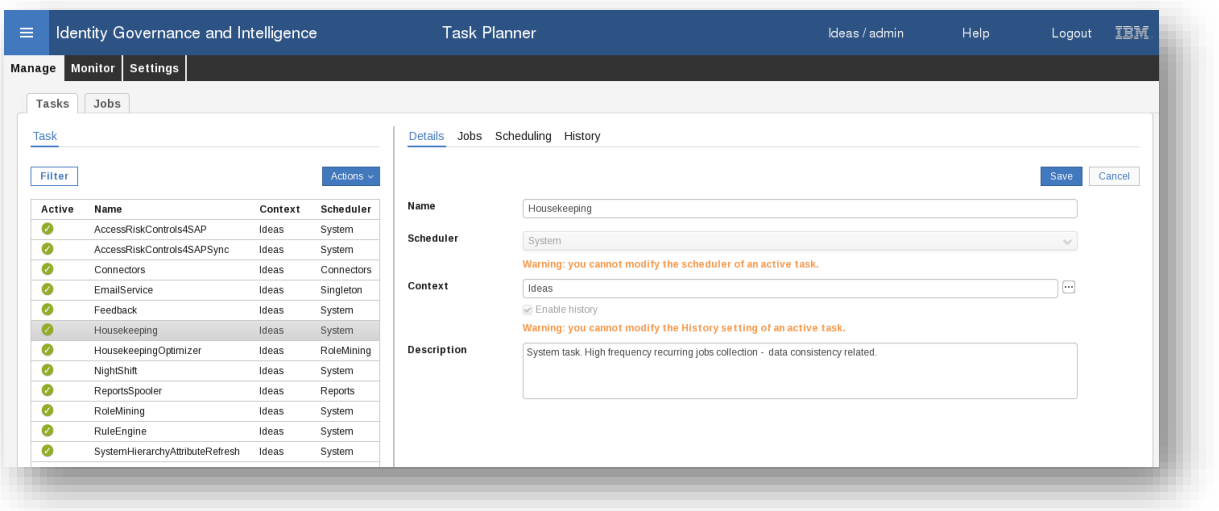


Figure 4: Mapping Task to Scheduler

Each Task is a group of jobs. Referring to the *Jobs* tab will provide a list of the jobs associated with a specific task and, more importantly, the order they are executed. The Task and Scheduler configuration which is provided by default in the Identity Governance product is sized for small to medium environments. Enterprise environments may need to adjust the Task and Scheduler association by moving Tasks to another scheduler, for example. The Rule Engine task might perform better if it is not sharing a scheduler with another Task that is long running, or CPU or IO intensive.

To move a Task to another scheduler, select it from the left frame Task Planner → Manage → Tasks → Task → Actions → Stop. Select the task from the left frame again, and under the *Details* tab in the right frame, choose the drop-down menu in the Scheduler bar. The scheduler can now be changed in the right frame. Save the changes and start the task. It is not necessary to restart the Identity Governance service. **It is important to note that if a Task is moved to another scheduler, it must be moved back to the original scheduler before Fixpacks or upgrades can be applied. The resulting upgrade or the application of a Fixpack may fail if a system Task is not located on its original Scheduler.**

Because the tasks cannot be monitored at the thread level, using the task history can be beneficial to determine when the task last ran, and the status for that run. More importantly, this history can provide information on how long it took to complete a previous run. This is important to know when the task scheduling needs to be adjusted. A task should not be scheduled to run in less time than it actually takes to complete the activity. This could result in blocked jobs in other tasks. Collecting history is not enabled by default. To enable it, use the method described above to stop

the task. In the *Details* tab, click the check box next to Enable History, then the Save button. Start the task in the left frame. It is not necessary to restart the Identity Governance service.

Certain tasks, such as the SAP related tasks, should not be run more often than every 1-2 days. Separation of Duty analysis, for example, can take a long time and consume significant processing power. By nature, the Separation of Duty analysis is computationally intensive. Using the history of a given task, and the built-in performance monitoring capabilities of the Administration Console, one can determine the time (from the history) and CPU requirements (from the Console) for an operation. This can help plan the frequency of such operations.

Although it is not possible to create additional schedulers beyond the six which are provided by default, the customer can use the Custom Tasks Scheduler to address specific needs. Such customization should be managed with care to avoid performance issues during runtime. It is recommended to enable the history of the Custom Tasks Scheduler when tasks are added to track the run time of the tasks within. The history can be disabled once the customer understands the run time and resource consumption for each task, adjusting the frequency if necessary.

5. Improving Rule Engine Concurrency

Reconciliation activities can be initiated through the Enterprise Connector module, which is accessible through the Administration Console. This activity proceeds in three phases. As an example, the data can be fed into the connector interface via a .csv file, which is then written into a connector table. Secondly, the connector will compare the record from the target (the .csv file) against the data in the Identity Governance system. When the connector determines that changes or updates are necessary, an event is created in the target event queue. Thirdly, the Rule Engine will begin processing those events in the target event queue. The second and third phases overlap. That is, the connector may still be creating events in the event queue, while the Rule Engine has already begun processing them.

Beginning in Version 5.2, the Rule Engine is now multi-threaded. As shown in Figure 2, there are five queues which are serviced by the Rule Engine. Four queues (the IN, TARGET, OUT, and INTERNAL queues) now have multiple threads to process events. The number of worker threads for each queue is now configurable, whereby the value **threadNumber** can be set from 1 to 10 threads. The optimal value is also the default value of 3 threads per queue. The Rule Engine must be stopped for this value to be adjusted.

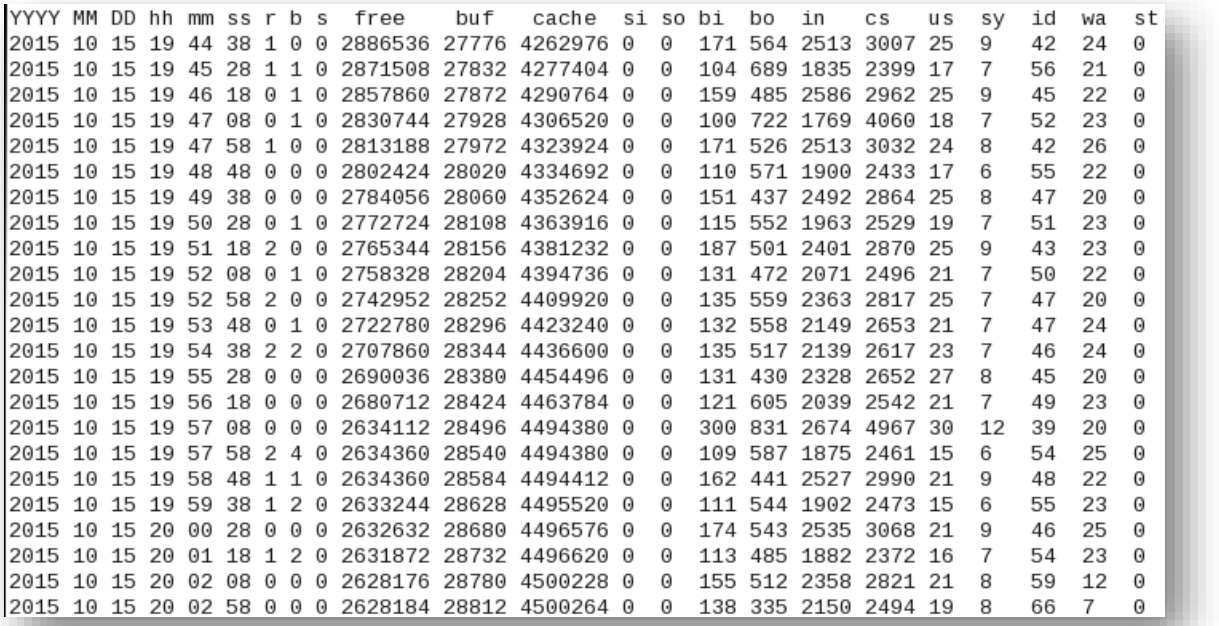
For event queues which service inbound or outbound events for multiple managed targets, it is recommended that there be at least one thread per target. The targets can be external systems such as an SAP R/3 system, an LDAP server, an Active Directory server, etc.

In Version 5.2.3 of the Identity Governance product, enhanced support was added for state coherency between the Governance product and a target endpoint. This support made it immediately clear if the two systems were not absolutely coherent in their reporting of provisioned accounts and permissions, irrespective of which system was the master. The support was accomplished with enhanced monitoring mechanisms in the insert, delete, and processing of changes to the database. These changes enable state coherency with a slight performance impact resulting in longer processing times and lower utilization of the DB and VA. In laboratory tests, the performance of a standard user account provisioning can be recovered (versus Version 5.2.2 results) by adding more threads to the OUT events queue. As mentioned before, closely monitor the CPU utilization when the number of Rule Engine threads is adjusted and increase this number slowly to avoid over-committing the CPU resources.

6. Reducing I/O Wait Time

With the introduction of the Identity Brokerage component in Version 5.2, comes the potential to incur I/O wait time due to multiple targets making changes to the Identity Governance data tier. Managing this wait time in the environment can translate to improved performance, higher transaction rates, and lower response times.

A simple *vmstat* output on the data tier can alert the system administrator to the need to tune the system to alleviate I/O wait time.



YYYY	MM	DD	hh	mm	ss	r	b	s	free	buf	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st
2015	10	15	19	44	38	1	0	0	2886536	27776	4262976	0	0	171	564	2513	3007	25	9	42	24	0
2015	10	15	19	45	28	1	1	0	2871508	27832	4277404	0	0	104	689	1835	2399	17	7	56	21	0
2015	10	15	19	46	18	0	1	0	2857860	27872	4290764	0	0	159	485	2586	2962	25	9	45	22	0
2015	10	15	19	47	08	0	1	0	2830744	27928	4306520	0	0	100	722	1769	4060	18	7	52	23	0
2015	10	15	19	47	58	1	0	0	2813188	27972	4323924	0	0	171	526	2513	3032	24	8	42	26	0
2015	10	15	19	48	48	0	0	0	2802424	28020	4334692	0	0	110	571	1900	2433	17	6	55	22	0
2015	10	15	19	49	38	0	0	0	2784056	28060	4352624	0	0	151	437	2492	2864	25	8	47	20	0
2015	10	15	19	50	28	0	1	0	2772724	28108	4363916	0	0	115	552	1963	2529	19	7	51	23	0
2015	10	15	19	51	18	2	0	0	2765344	28156	4381232	0	0	187	501	2401	2870	25	9	43	23	0
2015	10	15	19	52	08	0	1	0	2758328	28204	4394736	0	0	131	472	2071	2496	21	7	50	22	0
2015	10	15	19	52	58	2	0	0	2742952	28252	4409920	0	0	135	559	2363	2817	25	7	47	20	0
2015	10	15	19	53	48	0	1	0	2722780	28296	4423240	0	0	132	558	2149	2653	21	7	47	24	0
2015	10	15	19	54	38	2	2	0	2707860	28344	4436600	0	0	135	517	2139	2617	23	7	46	24	0
2015	10	15	19	55	28	0	0	0	2690036	28380	4454496	0	0	131	430	2328	2652	27	8	45	20	0
2015	10	15	19	56	18	0	0	0	2680712	28424	4463784	0	0	121	605	2039	2542	21	7	49	23	0
2015	10	15	19	57	08	0	0	0	2634112	28496	4494380	0	0	300	831	2674	4967	30	12	39	20	0
2015	10	15	19	57	58	2	4	0	2634360	28540	4494380	0	0	109	587	1875	2461	15	6	54	25	0
2015	10	15	19	58	48	1	1	0	2634360	28584	4494412	0	0	162	441	2527	2990	21	9	48	22	0
2015	10	15	19	59	38	1	2	0	2633244	28628	4495520	0	0	111	544	1902	2473	15	6	55	23	0
2015	10	15	20	00	28	0	0	0	2632632	28680	4496576	0	0	174	543	2535	3068	21	9	46	25	0
2015	10	15	20	01	18	1	2	0	2631872	28732	4496620	0	0	113	485	1882	2372	16	7	54	23	0
2015	10	15	20	02	08	0	0	0	2628176	28780	4500228	0	0	155	512	2358	2821	21	8	59	12	0
2015	10	15	20	02	58	0	0	0	2628184	28812	4500264	0	0	138	335	2150	2494	19	8	66	7	0

Figure 5: vmstat output

In the case shown in Figure 5, the database is reading and writing, although the blocks being written out dominate the I/O activity.

With buffer pool monitoring turned on, the database snapshots will reveal if the database is waiting on buffers to satisfy its I/O requirements. The hit ratio can provide information about the efficiency of the buffer pool usage. To calculate the hit ratio, refer to the snapshot for the values of logical to physical reads. In Figure 6, the buffer pool data is being compared.

Figure 6: Buffer pool information in DB snapshot

$$\frac{Logical\ reads - Physical\ reads}{Logical\ reads} * 100$$

Ideally, this hit ratio should be very close to 100%.

High write I/O, as seen through *iostat*, can also indicate the sort heap setting may be too small for the temporary space needed to a table or for performing database functions such as hash joins. Check for write I/O at the operating system level (using *iostat)* to determine the location of the write performance issue. If hash joins occur frequently, consider increasing the value of the sort heap parameter. The settings recommended in the Identity Governance documentation are suitable for the minimum VA (4-core). A more robust appliance may require more database resource.

High I/O wait time can also be caused by a very active DB transaction log on the data tier. To mitigate this situation, the transaction log should be moved to a separate, dedicated disk.

Although not entirely related to I/O wait time, the latency of the DB connection can also impact the performance of the system. It is recommended that the DB tier exist on the same subnet as the Identity Governance VA. The next best configuration is to house the DB in the same data center as the Identity Governance VA. In laboratory testing with a remote database in a separate data center, the increased latency has been observed to have a significant impact on UI responsiveness, for example. The same rules apply to the Brokerage component of the Identity Governance system. The Brokerage system should be as close to the DB as possible, in the same subnet or in the same data center. During provisioning and reconciliation exercises, the Brokerage component works closely with the DB during the event processing stages.

7. Bulk Load

Beginning V5.2.2, the Bulk Load engine is multi-threaded. A single bulk load file is now split into smaller segments and processed in parallel. In the past, customers could run two concurrent, but still singled-threaded, bulk loads to improve the transaction rate. With the improved concurrency of V5.2.2, customers should no longer need to run two concurrent bulk loads, although this option is still available. Care should be taken to ensure the CPU is not overcommitted when issuing more than 1 bulk load. The customer should start a single bulk load and ensure there are sufficient CPU resources before starting another concurrent load.

For long running bulk load jobs, it is advisable to disable data analysis tasks until the load is complete. As an example, the user can disable Role Mining, Data Exploration, System Hierarchy Refresh, or Separation of Duty jobs until the load is complete. This will reduce pressure on the CPUs of the Identity Governance system and avoid data analysis on an incomplete data set.

Lastly, in previous versions of Identity Governance, a performance enhancement involved moving the Housekeeping task to the Custom Tasks Scheduler. This allowed the housekeeping task, and thereby the Bulk Load engine, to have exclusive use of the two threads for that scheduler. This is no longer necessary in V5.2.2. In fact, if the Housekeeping task had been moved to the Custom Scheduler, it should be moved back to the System Scheduler to avoid problems with future upgrades. The tasks must be on the original schedulers to ensure that upgrades and migrations process successfully.

8. Collecting Java Core Dumps

A new feature as of Version 5.2.2 is the ability to collect java core dumps from the Identity Governance VA during runtime. Verbose garbage collection is not enabled by default in the VA but can be enabled via the CLI. From the root menu of the CLI, the administrator must navigate to igi → verbose_gc → enable. The Identity server must be restarted from the Administration Dashboard. In this way, verbose garbage collection can be enabled, a java core collected, and then verbose garbage collection can be disabled afterwards. Verbose garbage collection output is written to the verbosegc.log and does not affect the javacore. The javacore will list garbage collection statistics for a snapshot of time independent of whether verbose garbage collection is enabled. Enabling verbose garbage collection allows a complete record of all garbage collection activity to be captured, instead of a random short snapshot.

The customer can download the support file from the VA which will include a copy of the verbosegc.logs in *download_dir*/opt/ibm/wlp/usr/servers/igi/logs. Using the CLI, it is possible to enable, disable, and clear the log. To clear the log, the VA admin should enter the CLI path igi → logs → clear. An index will be presented. Option 4 will list the VA server logs. Choose the options to clear the verbosegc logs. Clearing the log does not require restarting the VA server. The same procedure can also be used for the Broker Application logs.

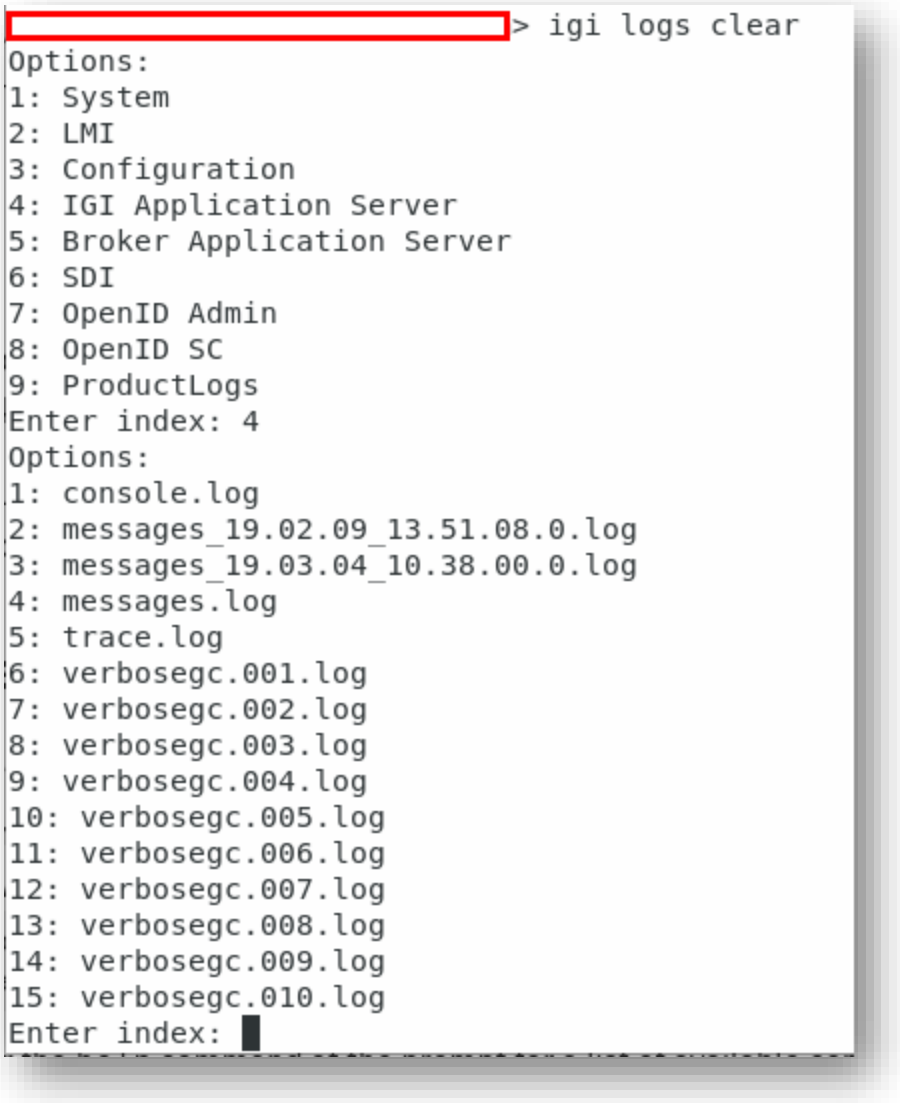


Figure 7: Clear VA Logs via CLI

To collect a java core file, the Administrator must navigate to Manage → Maintenance → Core/Heap Dumps on the Administration Dashboard. Choose the Generate option, then select which application server to collect.

The user will then click the check box to choose core dump, heap dump, or both. The heap dump will cause significant I/O for a short period of time, a condition which is most dramatic on the PostgreSQL environment where the DB is embedded. The customer should avoid collecting heap dumps unless instructed to do so by a field support technician. Additionally, such activities should be planned for a time when the VA is not busy with other CPU intensive activities.

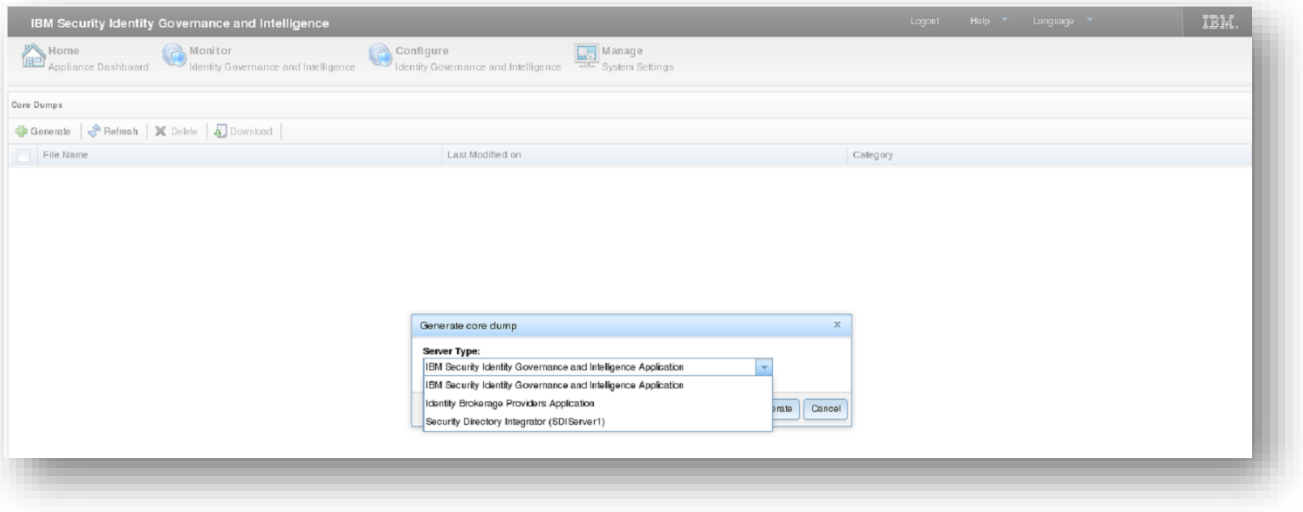


Figure 8: Generating a Java Core File

The appliance also includes an API method to generate a javacore, core dump, and/or heap dump.

Request

URL:

```
https://{appliance_hostname}:9443/v1/dmp_mgmt
```

Method:

```
POST
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Headers

Header	Description
`Content-Type:application/json`	Required for requests to the service.
`Authorization`	Basic Authentication header.
`Accept:application/json`	Required for requests to the service.

Request Example

```
POST https://{appliance_hostname}:9443/v1/dmp_mgmt
 POST_DATA:
 {
                generate_core_dump: "false",
                generate_heap_dump: "false",
                server_name: "igi"
 }
```

Response

Code: 201

Created

Figure 9: API to Generate Java Cores, Core Dumps, and Heap Dumps

The following remote command will cause a javacore to be created on the appliance for the IGI application server:

curl -s -k --user admin:*admin_password* -H "Accept: application/json" --data '{generate_core_dump: "false", generate_heap_dump: "false", server_name: "igi"}' POST https://*appliance_hostname*:9443/v1/dmp_mgmt

The javacore is created by default, and if the user requires a core dump or heap dump, the value can be set to "true". Once created, the user can then use the Administration Dashboard to list/download the core or heap dumps or use another API. The API definitions can be found from the Administration Dashboard by navigating to Help → Web Services.

9. PostgreSQL Database

In Version 5.2.2, Identity Governance introduced support for PostgreSQL. If the database type is set to PostgreSQL during the appliance installation, the database configuration is created automatically within the VA. The actual PostgreSQL database can either be truly embedded within the VA, or in an external tier where the database data is accessible via NFS mount. Both configurations have performance implications, but in both cases the actual DB administration runs inside the VA.

9.1 Embedded PostgreSQL Database

An embedded database is a desirable configuration for simple Proof of Concept or demo situations, because administration is low, and a hardware tier is eliminated. The PostgreSQL infrastructure itself has a low footprint requirement which makes it ideal for embedding in the VA. An embedded PostgreSQL database environment requires higher resource consumption than the standard external DB2 database, making it critical to increase memory and CPU allocation to ensure a stable operation of the environment. When the database is co-resident in the VA, the CPU and memory resources will be taxed additionally to provide services to the Identity Governance processes, as well as the database management processes. In laboratory tests, the CPU requirements on the VA are 2 to 3 times higher when running with PostgreSQL, versus the combined requirements of a VA and DB running with DB2. The additional memory and CPU requirements are most important in the PostgreSQL cluster scenario when data replication is enabled. Despite additional memory and CPU, the performance of this environment also falls behind that of DB2. At this time, PostgreSQL is not recommended for mission-critical environments, production, or deployments where performance requirements are high

In laboratory tests of a single VA configuration (not clustered), the mix of CPU utilization for Identity Governance processes and DB processes varies by operation. For reconciliation activities, the CPU utilization is evenly split between the Identity Governance processes (45%) and the PostgreSQL database processes (45%). For provisioning activities, the Identity Governance consumes more than half of the available CPU resources (56%) compared to the PostgreSQL DB processes (which consumes around 35%). If the VA configuration includes an embedded database, the CPU resources of the VA will need to be increased by at least 45% to avoid CPU exhaustion.

Beyond simplified administration, another advantage of the embedded PostgreSQL database can be found in the cluster configuration. When the VA is configured in a cluster, database replication can be enabled to provide an automatic High Availability (HA) scenario as a secondary node will house a copy of the database. Automatic failover capabilities are not supported, but the Administrator can use the Administration Console to manually promote the secondary node. To ensure failover capabilities are preserved, all nodes need to be configured with memory and CPU resources that match the primary node. As stated previously, additional resources are required to handle the processing requirements. Although data replication is a desirable feature, it comes at a cost. When data replication is enabled, scaling is problematic on a 2-node cluster. As an example, in laboratory tests of a reconciliation in a cluster the data replication feature requires 1.4X the processing power compared to the same operation without data replication. If data replication is enabled, it is especially important that the nodes participating in the cluster be on the same subnet to avoid latency concerns.

9.2 NFS Mounted PostgreSQL Database

The first thing to know is Identity Governance data replication services are not available if the PostgreSQL database is moved to an external mount point on an NFS server. In this case, it is left to the customer to implement HA on the NFS environment. As previously mentioned, best practices for a data tier of any type recommend the database be kept close to the Identity Governance server, within the same subnet if possible. During initial deployment of the VA, the database should be moved to the NFS prior to loading the data. As the database is being moved to the NFS, the database server is not running. Any Identity Governance transactions requiring database access will be stalled. In laboratory tests, moving an empty PostgreSQL database takes 2-3 minutes. A database of 120K users and 160K entitlements took 30-40 minutes.

If a cluster is moved to an NFS mount configuration, the slave database will disappear, and the cluster synchronization will instruct all member nodes to begin using the NFS definition. There is currently no method to migrate back to an embedded database configuration, once the database has been moved to an NFS mount. To return to an internal version of the PostgreSQL database,

the administrator must use backup/restore procedures. Refer the IGI product documentation for instructions.

The second important thing to know about an NFS mounted PostgreSQL DB is that the database processes are still running on the Identity Governance VA. Although the data is now housed externally, the resources required to manage the data (memory and CPU) will remain high on the VA.

The Identity Governance VA uses NFS3. The default options are seen below. Additional options can be set when the mount point is created through the Administration Console.

```
(rw,relatime,vers=3,rsize=32768,wsize=32768,namlen=255,hard,nolock,proto=udp,
port=65535,timeo=7,retrans=3,sec=sys,mountport=65535,mountproto=,local_lock=a
ll,addr=9.xxx.xxx.xxx)
```

The VA network interface will rely on the settings of the hardware on which it is running. The administrator is advised to inspect the Hypervisor of the underlying hardware to determine the NFS settings for Auto-Negotiation and the adapter speed. Auto negotiation should be turned on if the transmission rates are not known for the adapters on the end points, or the intermediate routers, switches, and hubs.

10. User Interface Dashboards

The User Interface (UI) has been updated in Version 5.2.2 with a new framework and new default options. One of those default options is an improved dashboard design for three personas: Employee, User Manager, and Application Manager. If a user logs into the Service Center with any of these personas, the dashboard presented to him will include several panels representing content loaded especially for him. User and Application managers might see access requests awaiting approval, delegation requests in process, or alerts from various applications. An employee user might see the status of a request for access, for example. This dashboard can be further customized to include additional content, swap out content, or remove content altogether. While these real time dashboards are desirable, they generate load on the VA server and the database. At times when the VA is lightly utilized, this additional CPU consumption may go unnoticed. However, at peak hours, the login load may interfere with other jobs, or the user may experience delayed login times or intermittent failures.

To mitigate the pressure on the VA and the database, the Administrator can disable the dashboard panels for the users with the following procedure. There are two steps to disabling the

dashboards. In the first step, the Administrator will login to the Administration Console and navigate to the Access Governance Core → Settings. The Administrator will need to check the boxes next to “Disable certification dashboards” and “Disable Access Requests dashboards”. Save the changes. This change will remove the selected panel from the dashboard at login time.

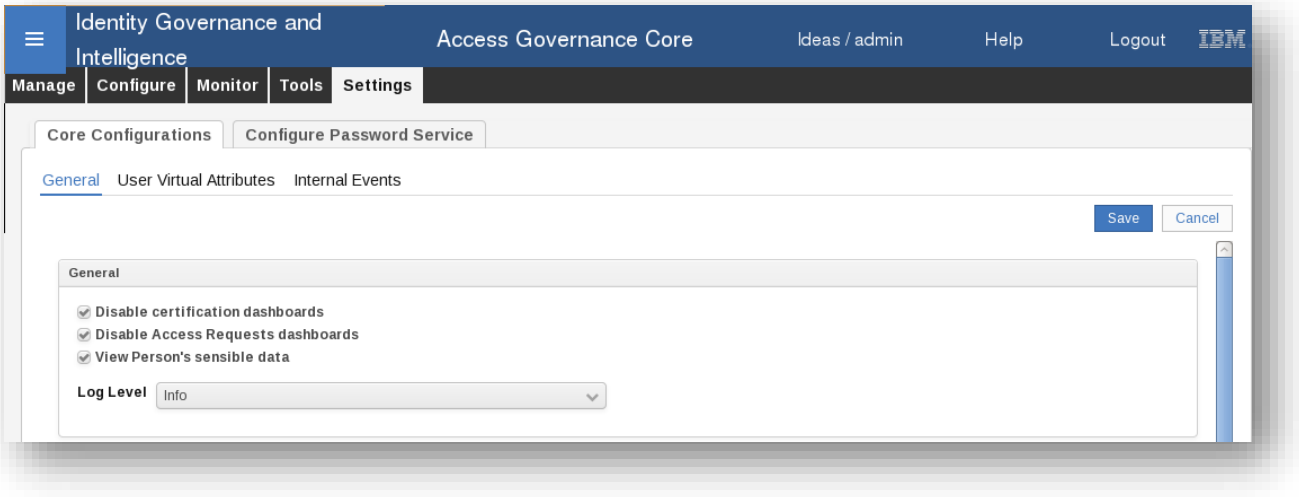


Figure 10: Disable dashboards

In the second step, the Administrator will navigate to the Report Designer Module → Configure → Assignment → Entitlement → Report Dashboard. In the left frame, set the Type to *Business Role* and click Search. In the list of roles which appear in the left frame, check the box next to User Manager. The right frame will be updated with the panels which will be displayed for this user persona. Check the boxes next to the dashboards to remove, then select Actions → Remove. This action will ensure the reports associated with the User Manager persona are not automatically generated each time that persona logs in.

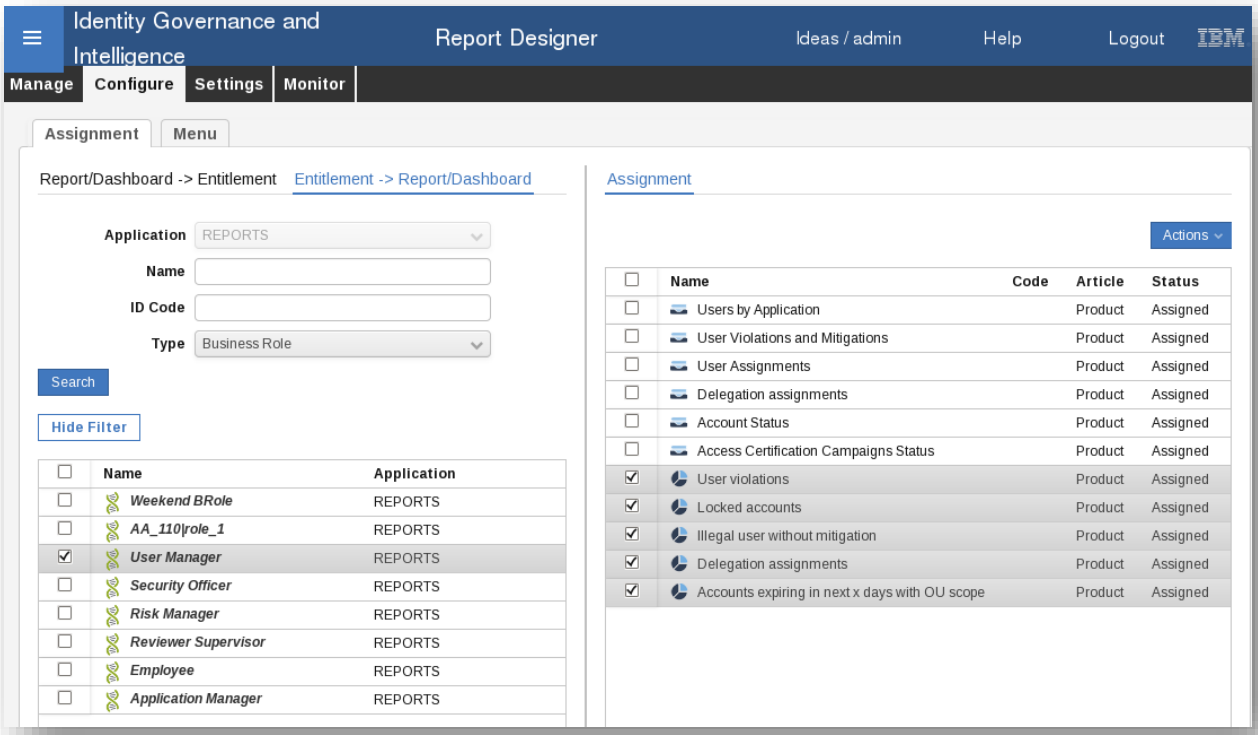


Figure 11: Remove Reports

Another method to improve the scaling of the UI is related to general best practices for a long running DB of any type. After sustained utilization of the UI, the table statistics and data access statistics should be updated. This reorganization should translate to reduced response times for frequently used data.

11. Improving Access Request Module Response Time

When the user logs into the Access Request Module, the query to populate the first landing page could delay the page load if there are many entries in the View Requests tab.

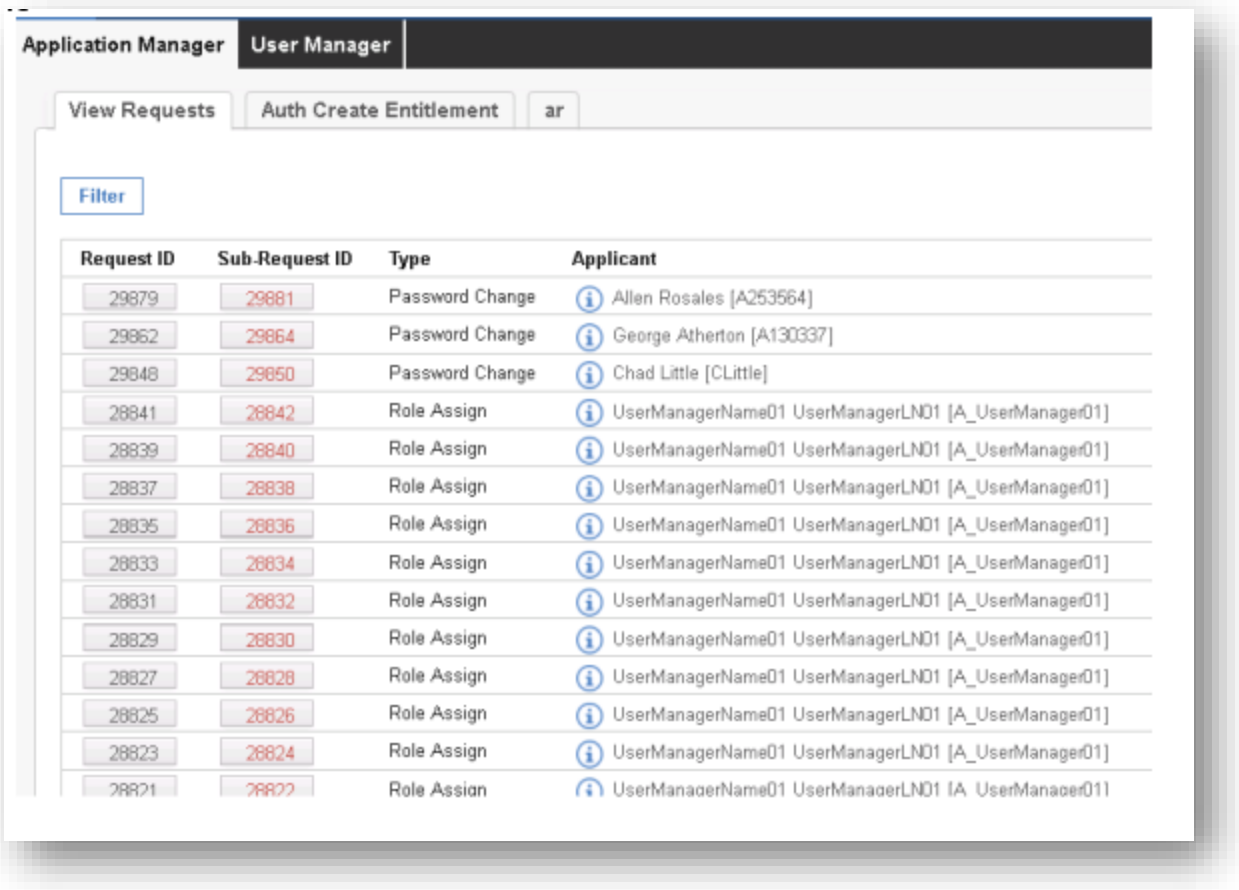


Figure 12: Default Login to ARM

These entries may, or may not, be of interest to the user upon first access. To mitigate the performance of first access, the first tab containing this information can be moved. That is, the administrator can customize the order in which the tabs appear to the users, delaying the query to populate the panel until the user actually chooses to see this data.

To reorder the tabs, the administrator can navigate to Process Designer → Manage → Process. Select the workflow in the left panel, then Actions → Maintenance.

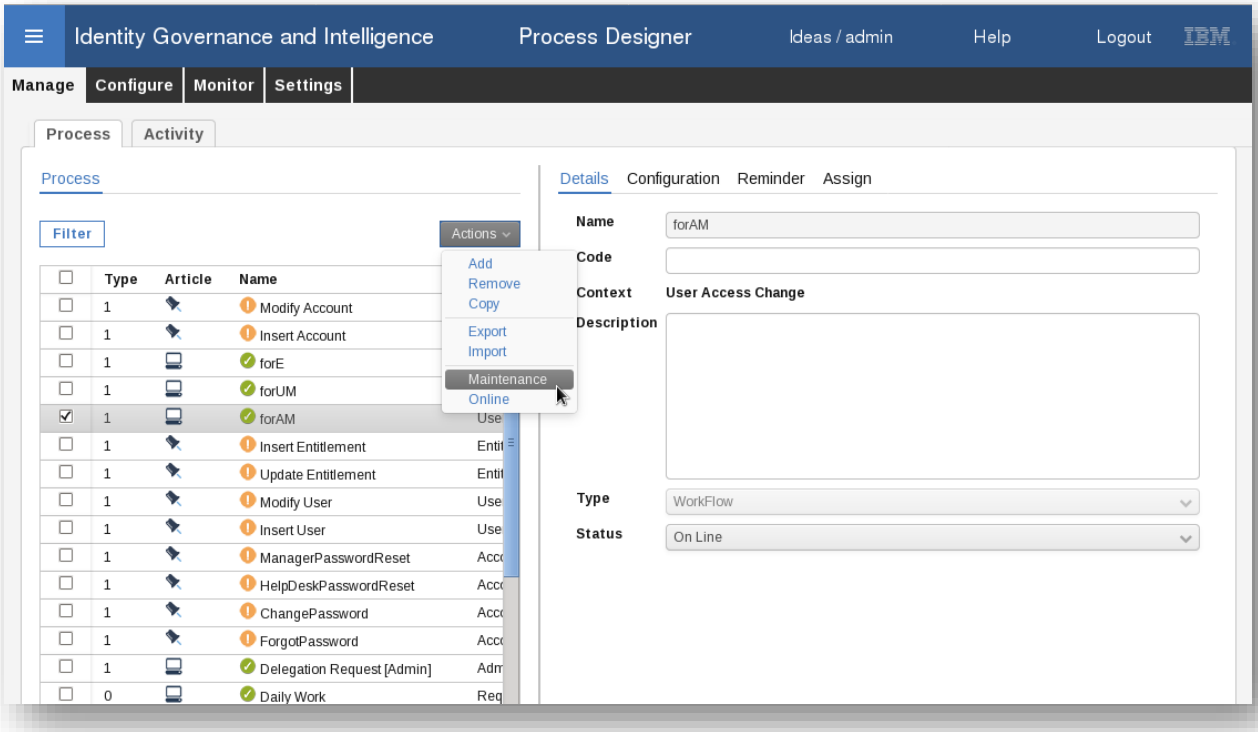


Figure 13: Select Workflow

The right panel will update for the given workflow. Navigate to Assign and select the Application. A new panel will appear on the right. Highlight a tab and then move up or down. The recommendation is to move the “View Requests” tab away from the top of the list. By moving another tab up (and to the left on the actual login page as the first tab), the panel that shows less data when the user navigates to this screen.

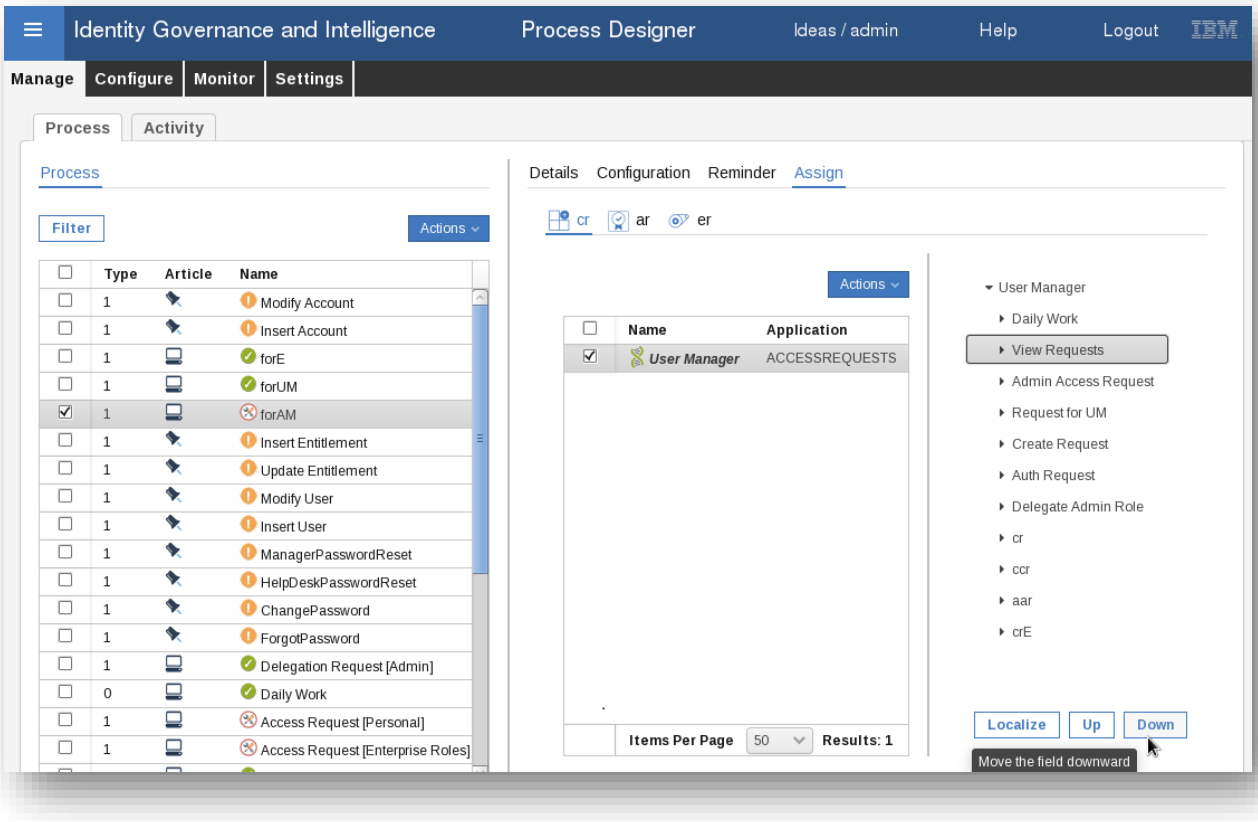


Figure 14: Move View Request Tab Down

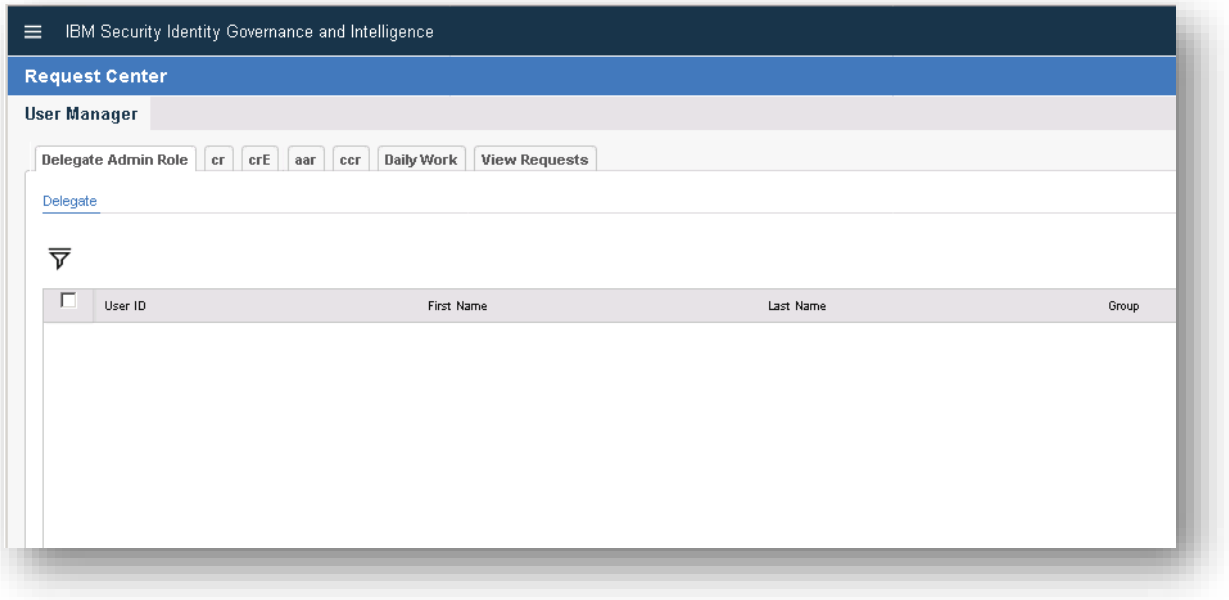


Figure 15: Example of Initial Page with No Data

12. Improving Access Certifier Module Response Time

In 5.2.4 FP1, laboratory testing indicated the response times in Access Certifier module can be improved of 10-15%, by adding new indexes to tables Employment_Review and Employment_Reviewer in the **igacore** schema.

- "create index igacore.emp_rev_att_person on igacore.employment_review(ATTESTATION,PERSON) collect detailed statistics "
- "create index igacore.emp_rever_cert_first on igacore.employment_reviewer(CERT_FIRST_OWNER,EMPLOYMENT_REVIEW) collect detailed statistics"
- "create index igacore.emp_rev_att_signoff_revstate on igacore.employment_review(ATTESTATION,SIGNED_OFF,REVIEW_STATE) collect detailed statistics "

13. UI Response Time at Application Server Restart

In laboratory testing, connections to the IGI User Interfaces are slow after a WAS restart and may be due to initial caching. The response time to load dashboards or to connect to a module,

such as Access Request Module or Access Certifier, suffer on the connection following the restart. Avoid applying stressful loads immediately after the WAS restart, then use an initial session login to the IGI application to repopulate the cache.

14. The Internal Security Directory Integrator

In Version 5.2.2, there are several important updates to the internal Security Directory Integrator (SDI) service. First, the customer can now create multiple SDI instances and secondly, the internal SDI can be used for general Directory Integrator services with the Identity Life Cycle Management and the Identity Brokerage. That is, there is no longer a requirement to use an external Tivoli Directory Integrator for operations which require Brokerage services.

The customer will use multiple SDIs if there are adapter conflicts amongst the IBM Security Identity Adapters resident on the Identity Governance configuration. These SDI instances also provide high availability services in cluster environments. In Version 5.2.2, up to a maximum of 10 SDI instances can be configured. Each instance will have a heap associated with it which is configurable at creation time. Laboratory tests for multiple SDI instances indicate that a max heap setting of 1GB is sufficient to achieve performance and resource requirements, without putting too much pressure on the memory subsystem. In laboratory tests, each additional SDI translates to 6% additional CPU consumption. In a system where there is abundant CPU available, this will not translate to a performance impact. However, a system with high CPU demands will likely see an impact for multiple SDI instances.

Using the single internal SDI, rather than an external TDI server, does not impact the performance of the system for Identity Life Cycle Management and the Identity Brokerage operations.

15. System Hierarchy Refresh

V5.2.2 contains significant improvements to the hierarchy calculations. While improved, the nature of a hierarchy refresh might mean extended run time to complete the entire scan. It is important to observe, via the History of the SystemHierarchyAttributeRefresh Task, that the job is not scheduled to run too frequently. If it takes 5 hours to run, the schedule should not be set for less than that time. The default time for this Task is 3 hours, an interval which may be too short for an Enterprise Governance environment. To adjust the frequency, stop the

SystemHierarchyAttributeRefresh Task, adjust the frequency to a larger value, save the setting, then start the Task. It is not necessary to restart the Identity Governance process.

16. Enabling FIPS and SSL

In Version 5.2.3, the Identity Governance product shipped support for Federal Information Processing Standards (FIPS). The default deployment of an Identity Governance VA is without FIPS support, but when FIPS is enabled, the environment must also include SSL enabled DB connections. It is not possible to migrate a VA from a previously non-FIPS state to a FIPS state. To achieve FIPS support, the VA must be installed anew. Prior to connecting a VA with FIPS enabled to an existing DB, the DB connections must be hardened with SSL support. Refer to the online Knowledge Center documentation for support and migration procedures.

A FIPS+SSL environment will show a reduction in performance versus the standard no FIPS deployment. These performance impacts are most likely to be seen when interacting with the User Interface (UI). To isolate the effects of FIPS enablement, laboratory tests were conducted for basic Service Center Logins, Self-Care Change Password functions, and navigation to the Access Certifier and Access Request modules. In each case, comparisons were made with 100 concurrent users between a FIPS environment and a standard deployment without FIPS support. The results were collected on a VA with 6 cores (rather than the default 4 cores) to ensure that additional CPU utilization associated with FIPS calculations would not exhaust the CPU resources. With FIPS enabled, the CPU resource requirements of the VA per transaction increased by an additional 10% - 30%. The database resource requirements per transaction remained relatively the same.

17. Clearing the Event Queues

The Rule Engine is responsible for processing events in its 5 work queues. Searching the queues is a normal part of the operation of this Task, as well as updating and inserting events. If the queues contain many entries, the time it takes to search, update, or insert into the queue will cause delays in event processing. While 100, 10,000, or even half a million events may not affect event processing performance, letting the queues grow beyond this might. A useful maintenance strategy is to clear the event queues periodically to reduce the search time. How often to clear the queues is dependent upon the activities of the Identity Governance environment and how often events are created. The customer can use the filter option to list only those events which are successfully completed and remove them.

For example, events in the Access Governance Core queues should be inspected, AGC → Monitor.

1. Scheduled Tasks
2. TARGET inbound – Account events
3. TARGET inbound – Access events
4. OUT events
5. IN – User events
6. IN – Org. Unit events
7. INTERNAL events

18. Enabling SNMP for Performance Monitoring

The Identity Governance product offers monitoring from the Administration Dashboard. The user may navigate to Monitor → Monitoring. There are three options to view common performance statistics: Memory, CPU, and Storage. The Memory graph will provide a view of memory statistics for 1, 3, 7 or 30 days. In the same way, the CPU utilization and VA Storage can be viewed in graph format.

There is an additional option for performance monitoring available. Performance of the Identity Governance product can be monitored via the Simple Network Management Protocol (SNMP). Using this popular tool, one can query the VA for standard performance statistics. To configure the VA for an SNMP connection, log onto the Administration Console → Monitor → Monitoring → SNMP Monitoring. Check the radio button next to SNMP Monitoring and click Reconfigure.

Figure 16: Enable Monitoring

A pop-up window will appear to specify the SNMP version, port, and the community name. Fill in the values and click Save Configuration.

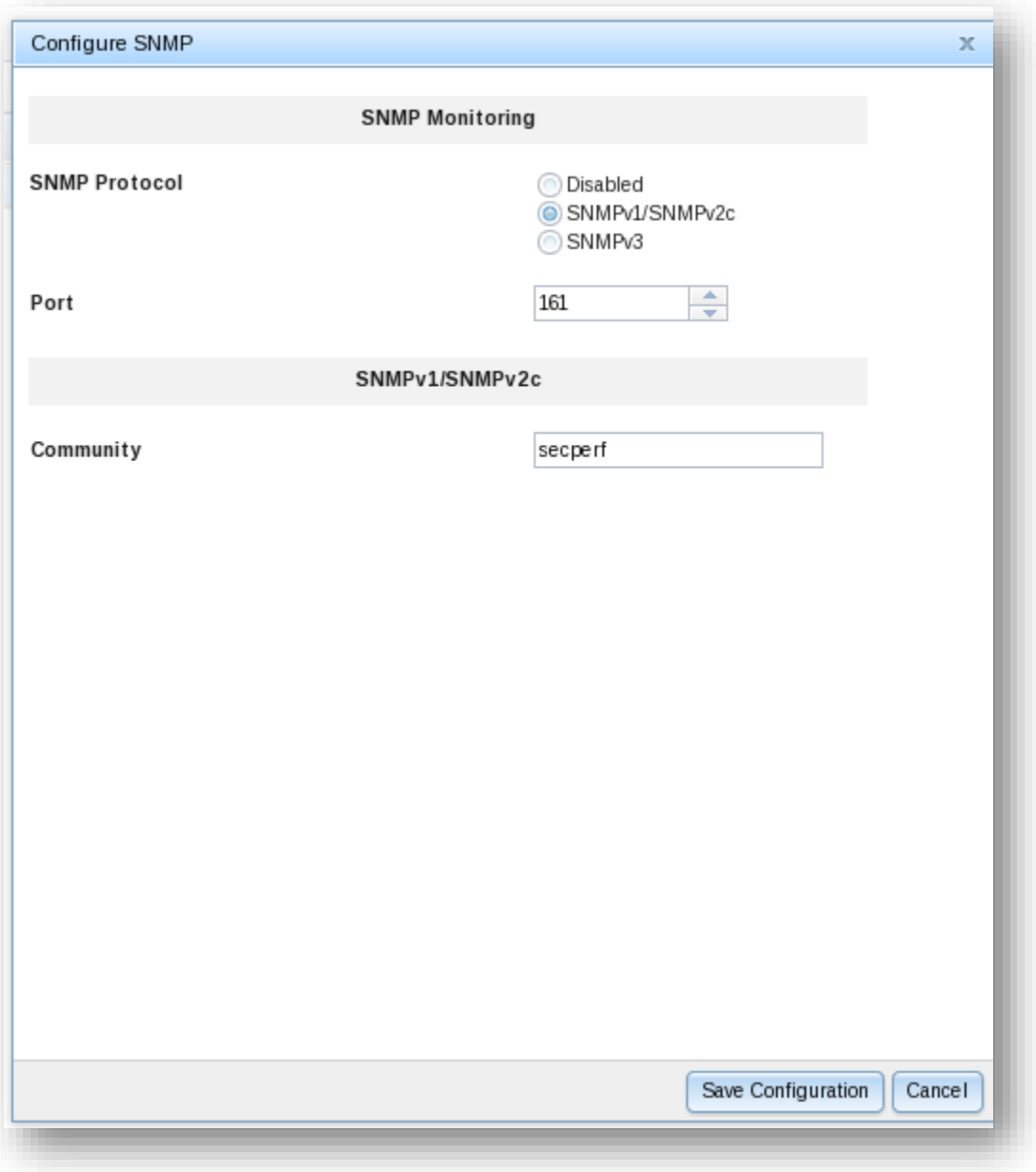


Figure 17: Set SNMP Version and Community

The SNMP processes will be started, but the Liberty processes for the VA do not require a restart. To enable alerts for SNMP, navigate to Manage → System Settings → System Audit Events.

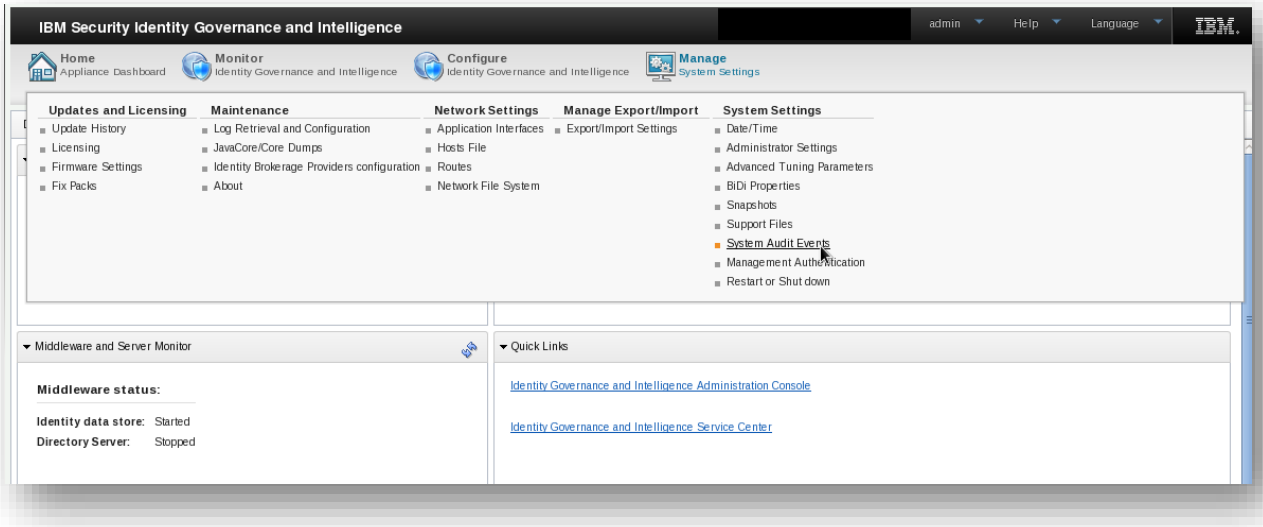


Figure 18: System Audit Events

Add an SNMP object by clicking New → SNMP.

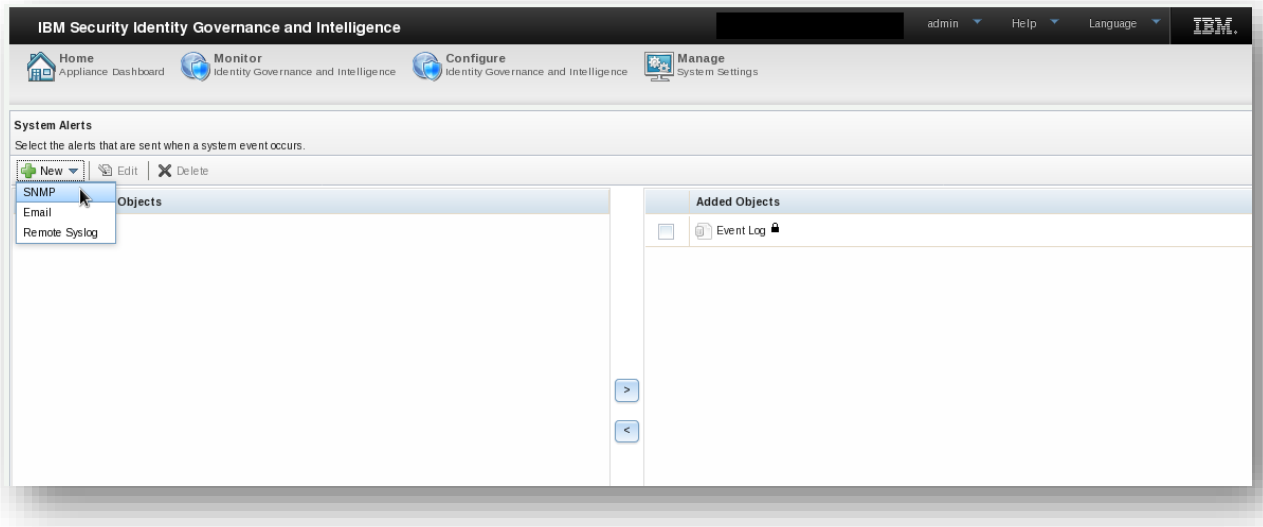


Figure 19: Add SNMP Object

A pop-up window will appear. Fill the information to name the object, along with the version, the IP address, the port, and the community. Save the configuration.

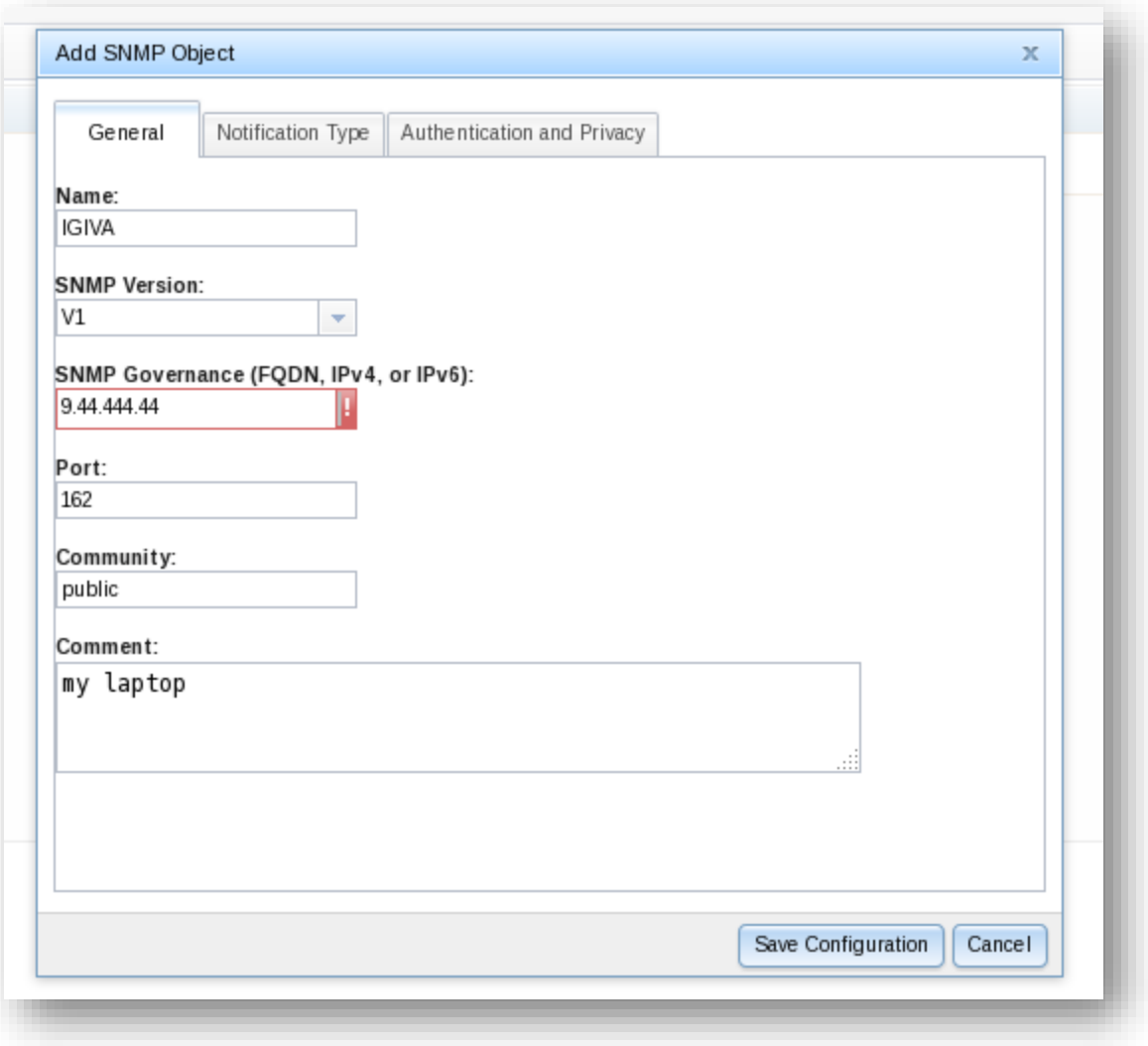


Figure 20: Configure Alerts

Use “>” button in the center of the window to add this to the Added Objects panel.

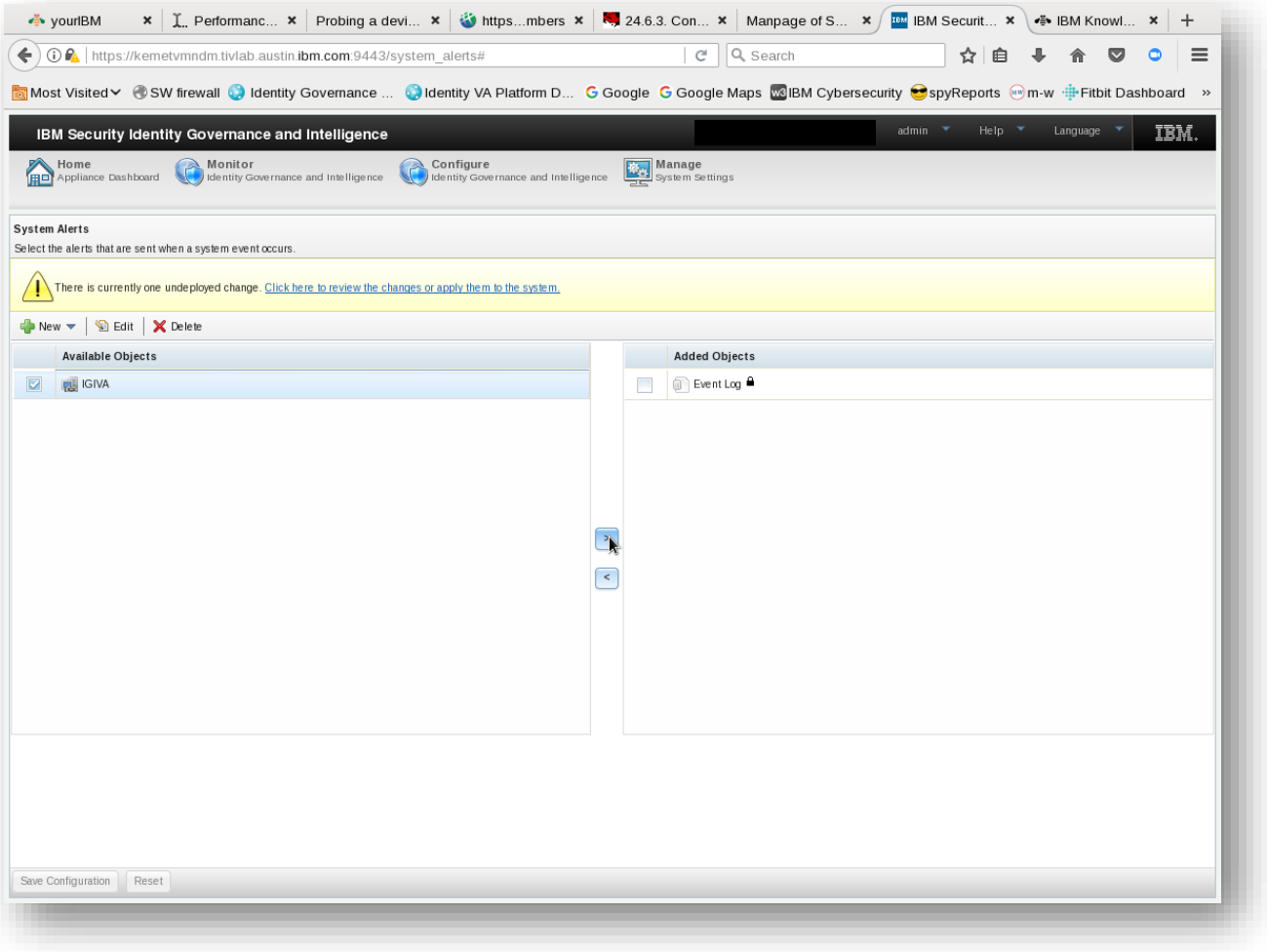


Figure 21: Deploy Changes to Alerts

Click Save Configuration. Deploy the changes.

Once the system is configured, it can be tested/accessed with a simple command such as

```
snmpwalk -v1 -c community hostname
```

To begin with, the *snmpwalk* is a good way to determine the object labels available in your VA. Redirect the output from the command above to a local file and reference it to determine the labels for polling. The values in the MIB variables are counters, so to determine input or output rates (for example), one will need two poll cycles to figure the difference between them.

With these labels, one can then write custom scripts to poll this information directly from the system. As an example, the following command will collect CPU utilization information on the VA every 10 seconds.

```
while ((1)); do snmpwalk –v1 –c community hostname HOST-RESOURCES-
MIB::hrProcessorLoad ; sleep 10;
```

Online man pages for SNMP caution against polling too often (1 second) because this can artificially inflate the CPU utilization.

IO behavior can be collected. In the example below, inbound traffic on the network object at index 1 is polled several seconds apart with the following command.

```
snmpwalk -v1 -c community hostname | grep ifInOctets | grep "\.1 "
```

Response #1: 489831639

Response #2: 490500051

The customer will need to do the appropriate math to determine the rate of network traffic based on the polling period, the duplex setting of the network, and the speed of the network interface.

Users can also collect per-process CPU utilization statistics from the *snmpwalk* information. The MIB data will report the process index. The process almost always found at index 1 on a Unix system is **init**. Looking at a sample of the *snmpwalk* output, one can see the index of processes.

```
HOST-RESOURCES-MIB::hrSWRunIndex.1 = INTEGER: 1
HOST-RESOURCES-MIB::hrSWRunIndex.2 = INTEGER: 2
HOST-RESOURCES-MIB::hrSWRunIndex.3 = INTEGER: 3
HOST-RESOURCES-MIB::hrSWRunIndex.4 = INTEGER: 4
HOST-RESOURCES-MIB::hrSWRunIndex.5 = INTEGER: 5
HOST-RESOURCES-MIB::hrSWRunIndex.6 = INTEGER: 6
```

```
HOST-RESOURCES-MIB::hrSWRunIndex.7 = INTEGER: 7
HOST-RESOURCES-MIB::hrSWRunIndex.8 = INTEGER: 8
HOST-RESOURCES-MIB::hrSWRunIndex.9 = INTEGER: 9
HOST-RESOURCES-MIB::hrSWRunIndex.10 = INTEGER: 10
HOST-RESOURCES-MIB::hrSWRunIndex.11 = INTEGER: 11
HOST-RESOURCES-MIB::hrSWRunIndex.12 = INTEGER: 12
```

The index can be used to find the name of the process by looking at the hrSWRunName object.

```
HOST-RESOURCES-MIB::hrSWRunName.1 = STRING: "init"
HOST-RESOURCES-MIB::hrSWRunName.2 = STRING: "kthreadd"
HOST-RESOURCES-MIB::hrSWRunName.3 = STRING: "migration/0"
HOST-RESOURCES-MIB::hrSWRunName.4 = STRING: "ksoftirqd/0"
HOST-RESOURCES-MIB::hrSWRunName.5 = STRING: "migration/0"
HOST-RESOURCES-MIB::hrSWRunName.6 = STRING: "watchdog/0"
HOST-RESOURCES-MIB::hrSWRunName.7 = STRING: "migration/1"
HOST-RESOURCES-MIB::hrSWRunName.8 = STRING: "migration/1"
HOST-RESOURCES-MIB::hrSWRunName.9 = STRING: "ksoftirqd/1"
HOST-RESOURCES-MIB::hrSWRunName.10 = STRING: "watchdog/1"
HOST-RESOURCES-MIB::hrSWRunName.11 = STRING: "migration/2"
HOST-RESOURCES-MIB::hrSWRunName.12 = STRING: "migration/2"
```

The following command can be run a loop to collect the utilization of a single process over time. In this example, the process being examined is the **init** process (index 1).

```
snmpwalk -v1 -c community hostname | grep HOST-RESOURCES-MIB::hrSW |
grep "\.1 "
```

This is the output.

```
HOST-RESOURCES-MIB::hrSWRunIndex.1 = INTEGER: 1
HOST-RESOURCES-MIB::hrSWRunName.1 = STRING: "init"
```

```
HOST-RESOURCES-MIB::hrSWRunID.1 = OID: SNMPv2-SMI::zeroDotZero
HOST-RESOURCES-MIB::hrSWRunPath.1 = STRING: "init"
HOST-RESOURCES-MIB::hrSWRunParameters.1 = ""
HOST-RESOURCES-MIB::hrSWRunType.1 = INTEGER: application(4)
HOST-RESOURCES-MIB::hrSWRunStatus.1 = INTEGER: runnable(2)
HOST-RESOURCES-MIB::hrSWRunPerfCPU.1 = INTEGER: 111
HOST-RESOURCES-MIB::hrSWRunPerfMem.1 = INTEGER: 856 KBytes
```

In this example, the **init** process took 111 centi-seconds of the total system's CPU resources. Although memory utilization is available through the VA Administration Dashboard, the user might find the SNMP information useful for per-process memory utilization statistics as well.

Refer to the online documentation and man page(s) for definitions of the fields and labels, additional examples, and help for SNMP.

19. DB Connection Pool

The default number of connections per resource for the Identity Governance product is 30. For Enterprise Environments, medium sized environments with many targets, or environments expecting many concurrent user logins, this default setting may not be adequate. The customer may experience gradually deteriorating UI performance and/or sluggish response times. To determine if these problems are due to a low connection setting, the user should look at the WebSphere Liberty logs for the Identity Governance application server. Download the support files from Administration Console → Manage → System Settings → Support Files.

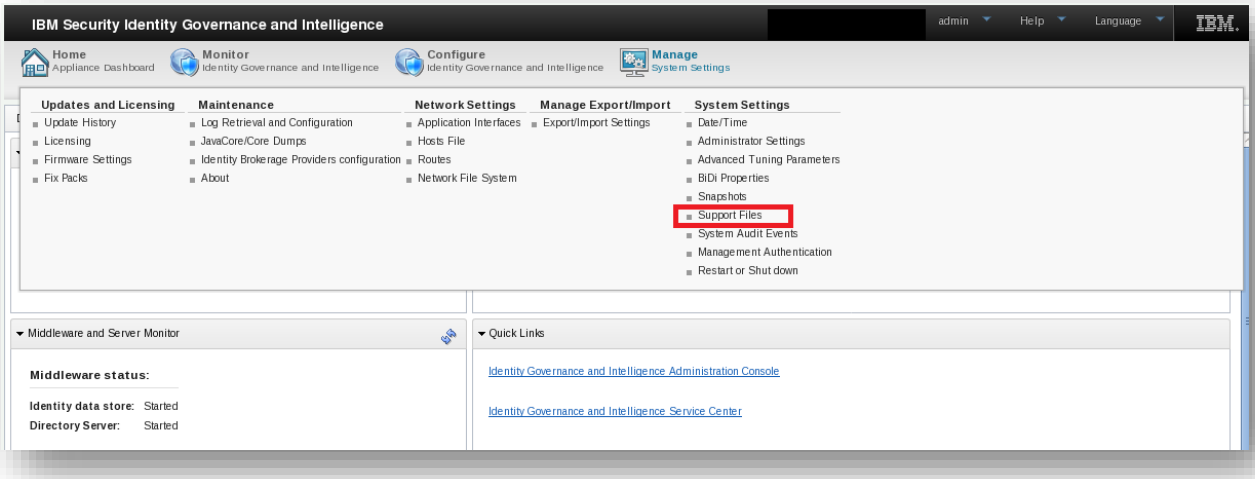


Figure 22: Downloading Support Files

In the log file at opt/ibm/wlp/usr/servers/igi/logs/messages.log, look for errors associated with connections not being available.

```
00000066 SystemErr          R       java.sql.SQLTransientConnectionException:
Connection not available, Timed out waiting for 180000
```

Such messages can also be found /opt/ibm/wlp/usr/servers/igi/logs/ffdc in the exception summaries.

```
JST com.ibm.websphere.ce.j2c.ConnectionWaitTimeoutException Max connections
reached 869
```

When there are no connections available, the Identity Governance core operations will also see errors. For example, event processing may encounter an error such as the one below.

```
ERROR AGC:? - openConnection failed

org.hibernate.exception.JDBCConnectionException: Cannot open connection
```

To avoid connection failures, the DB connection pool should be set high enough to accommodate the anticipate connection load. From the Administration Dashboard → Configure

→ Database Server Connection. Click the radio button next to the current Identity data store and click Reconfigure. In the pop-up window, go to the Connection Pool tab and edit the maximum number of connections. The Identity Governance application server must be restarted for the new connection setting to go into effect. A setting of 50 is a good first step, but some enterprise environments require as many as 100 connections. A good way to monitor the demand for connections is with a DB snapshot. As an example, in a DB2 snapshot search for the phrase "High water mark for connections"

```
# grep "High water mark for connections" ISIG-snapshot*.out
ISIG-snapshot1.out:High water mark for connections            = 31
ISIG-snapshot2.out:High water mark for connections            = 50
```

20. Multi-threaded Enterprise Connector

The ability to create multi-threaded enterprise connectors was added to the Governance product in Version 5.2.3. When an enterprise connector is created, the administrator may now specify the number of threads which will process the events for a given endpoint target. The default number of threads is 3, but this value can be tuned to a value in the range 1 – 10. To set this value, the administrator can edit the existing configuration of an enterprise connector or set it when the connector is newly created.

To set this value for an existing connector, the administrator must stop the connector, Enterprise Connectors (EC) → Manage → Connectors. Choose the connector from the left panel, then open the Global Config section in the right panel.

The attribute is called the "WRITE_TO_THREADS_NUMBER". When a connector is created, this value is not surfaced in the configuration.

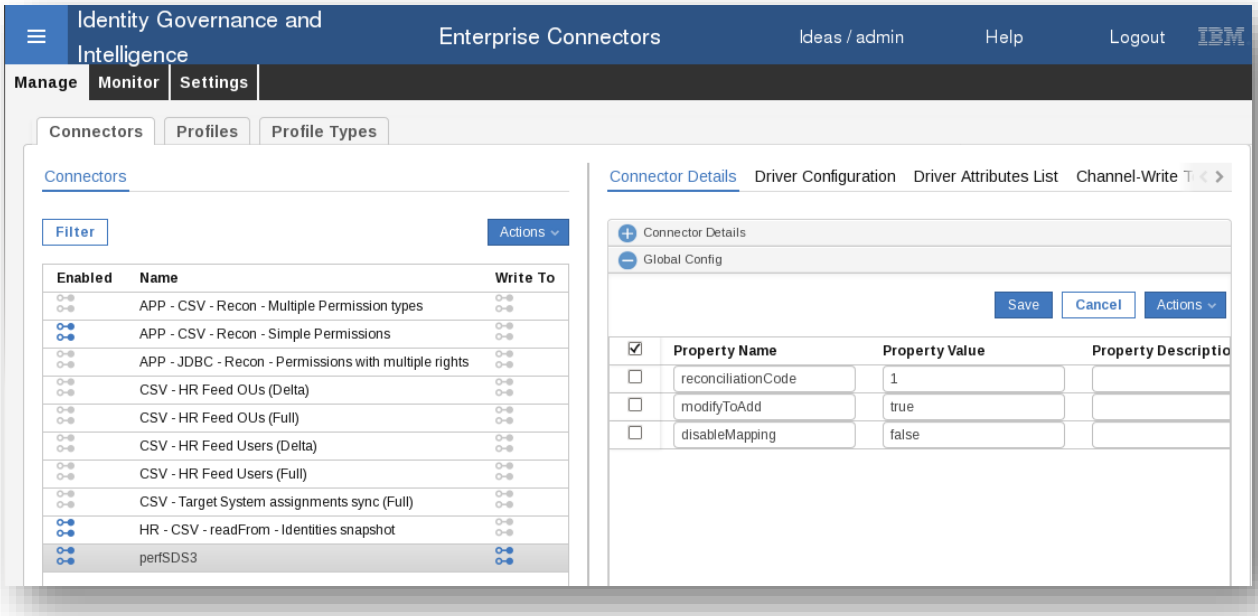


Figure 23: Global Configuration of Enterprise Connector

If the administrator wishes to alter this property, choose Actions → Add in the right panel. An empty field will be added to the panel. Enter WRITE_TO_THREADS_NUMBER in the *Property Name* field, the number of threads to set in the *Property Value* field, and an optional description in the *Property Description* field. Click Save.

In the same way, when a connector is newly created, this value can be set in the initial configuration. There is no need to stop/start/restart the Governance application server.

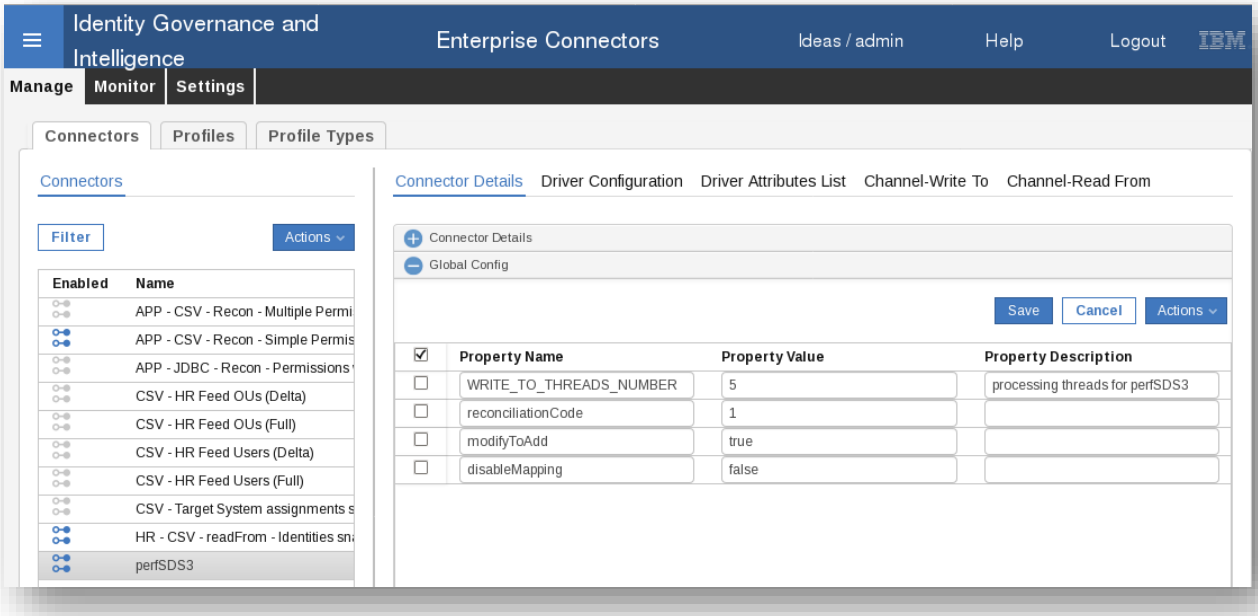


Figure 24: Setting WRITE_TO_THREADS_NUMBER

Although the default configuration already contains multi-threading (3), the concurrency can be increased for the processing of the events in the OUT queue. The tests used to demonstrate this feature separated the event creation from the event processing. Under ordinary (default) circumstances, events are created in the OUT queue while the Enterprise Connector threads are processing them (fulfillment). For the purposes of this test, and to demonstrate the precise effect of this feature, the two phases were separated into event creation and event processing. In laboratory tests with 3, 5, 7, and 9 threads, the event processing showed increasing raw throughput, and increased overall system efficiency (DB + Governance VA). For the system under test (laboratory environment) the peak for both throughput and efficiency was 9 threads.

Obviously, the DB and VA will see increased CPU utilization when the thread concurrency is increased. In the laboratory, going from 3 to 5 threads increased the both DB and VA CPU utilization by 3-4%. Going from 5 to 7 threads increased the DB CPU utilization by an additional 20+%. The VA CPU increased by only 7%. The utilization is effective flat for both the DB and VA going from 7 to 9 threads, with a slight raw throughput increase at 9 threads.

When invoking this feature in a production environment, there are three things to remember. The first consideration is a default processing environment will involve event creation and event processing occurring at the same time. CPU resources will need to be split across the two. Setting WRITE_TO_THREADS_NUMBER too high may choke the CPU or overrun the event

creation process. If there are no events to process, the multiple threads of the connector will go to waste. The second consideration is the other background tasks that may need CPU resources. Care should be taken; tune this setting slowly to avoid CPU starvation for other tasks within the VA and those which need DB services. Lastly, faster event processing will cause the endpoint target, the TDI, and the SDS to see increased CPU utilization. Although these increases were small in the laboratory environment, this may not be the case in the production environment where those machines/services may be housed on shared hardware where CPU is a premium resource.

21. Tcpdump

Support was added in V5.2.4 for collecting tcpdump results from the Governance appliance. The tool is available via the CLI tools → packet_tracing and is not enabled by default. Below is an example of accessing the tool and running a trace.

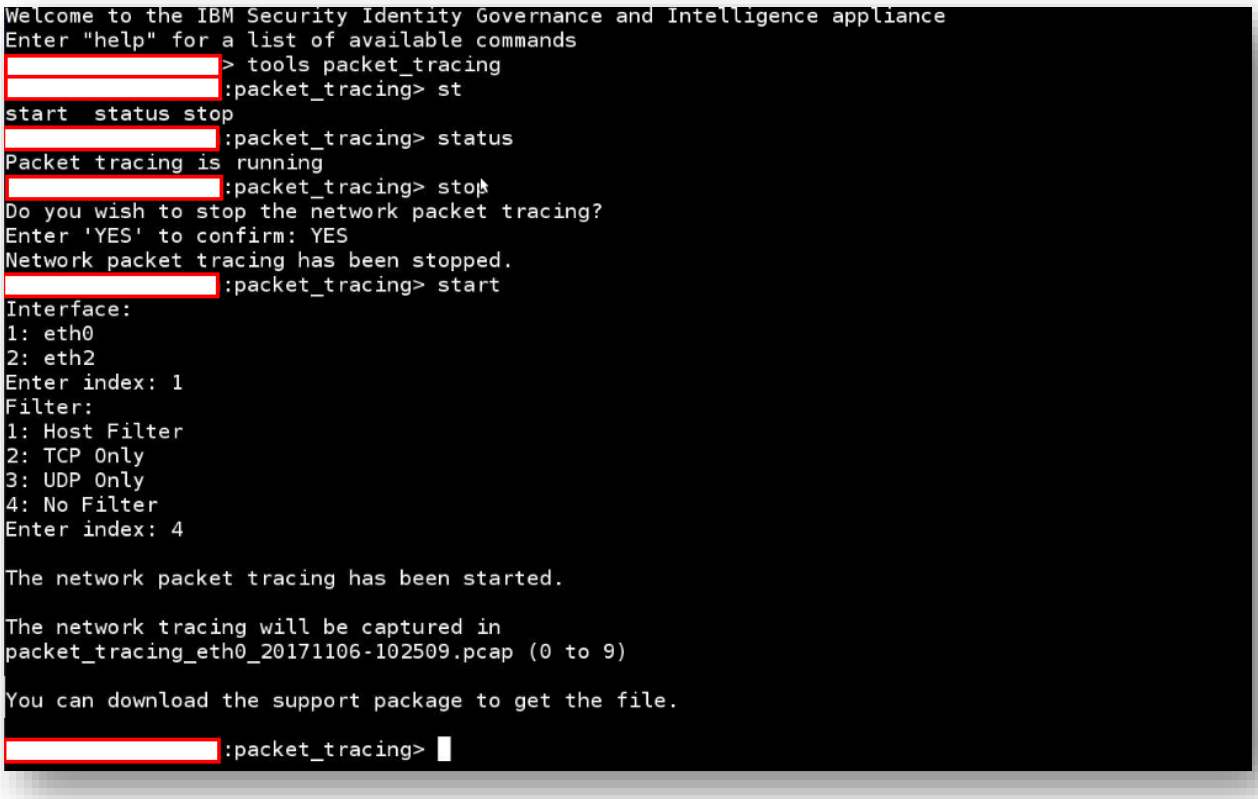


Figure 25: Tcpdump Example

22. Tuning the Directory Server

When Identity Brokerage services are invoked, a directory server is a mandatory part of the Governance configuration. To ensure this component does not become the bottleneck during reconciliation and fulfillment operations, it should be tuned by creating indexes on the following tables.

ou

erparent

erglobalid

eroperationnames

This tuning is especially important if the Multi-Threaded Enterprise Connector feature is invoked with threads higher than the default number (3). These additional threads will put pressure on the directory server which manifests as higher CPU consumption. In laboratory tests, with a well-tuned Governance system and multi-threaded enterprise connectors set to 5 threads, the directory server proved to be a bottleneck until these indexes were applied. Consult the documentation for creating indexes on the Security Directory Server:

https://www.ibm.com/support/knowledgecenter/en/SSVJJU_6.4.0/com.ibm.IBMDS.doc_6.4/c_tg_db2_indexes.html

23. Increasing the Heap Size

V5.2.4 provides a mechanism to increase the heap size to a max of 8GB. This step might be necessary when running large bulk loads, or launching many or particularly large hierarchies at the same time, etc. If the VA reports an error message that the heap has been exhausted, the following procedure can be used to increase the heap. From the CLI navigate to igi → jvm_heapsize → set_max_heapsize. The menu will offer an option to change the IGI application server heap or the Broker application server heap. The user may choose any size between 4096 MB and 8192 MB. A restart of the application server is necessary.

24. Resetting a Connector and Clearing Brokerage Data

As of V5.2.4, the Governance product offers a mechanism to reset a connector and clear the brokerage data for the target application. Refer to the Knowledge Center for prerequisites and

instructions. As stated in the Knowledge Center, this operation should be a last resort, and there are performance implications of using this procedure. The coherency between the Governance VA and the target must undergo a synchronization to align their states. This may result in a repopulation of the cached data which means CPU consumption on both the VA and the DB tier. Depending on the amount of data to be repopulated, this resynchronization could take a significant amount of time and should be scheduled during periods of low activity.

Prior to starting this procedure, data analytics should be turned off (Risk Scans, Role Mining, etc.) and resumed after the operation is complete. Additionally, it is advisable that database optimization techniques like *runstats* and *db2rbind* be executed when the operation is complete to ensure statistics are updated.

25. Deadlocking on Foreign Key Constraints

During testing of V5.2.4, the laboratory environment encountered a database deadlock during a performance test of a hierarchy build. Analysis of the condition revealed the cause to be foreign key constraint definitions that had no obvious function. To avoid this situation in future test cycles, the performance labs dropped all foreign key constraints using the following DB commands.

```
ALTER TABLE IGAQRZ.QRZ1_BLOB_TRIGGERS DROP FOREIGN KEY QRZ1_BLOB_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ1_CRON_TRIGGERS DROP FOREIGN KEY QRZ1_CRON_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ1_SIMPLE_TRIGGERS DROP FOREIGN KEY QRZ1_SIMPLE_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ1_SIMPROP_TRIGGERS DROP FOREIGN KEY QRZ1_SIMPROP_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ1_TRIGGERS DROP FOREIGN KEY QRZ1_TRIGGER_TO_JOBS_FK;
ALTER TABLE IGAQRZ.QRZ2_BLOB_TRIGGERS DROP FOREIGN KEY QRZ2_BLOB_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ2_CRON_TRIGGERS DROP FOREIGN KEY QRZ2_CRON_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ2_SIMPLE_TRIGGERS DROP FOREIGN KEY QRZ2_SIMPLE_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ2_SIMPROP_TRIGGERS DROP FOREIGN KEY QRZ2_SIMPROP_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ2_TRIGGERS DROP FOREIGN KEY QRZ2_TRIGGER_TO_JOBS_FK;
ALTER TABLE IGAQRZ.QRZ3_BLOB_TRIGGERS DROP FOREIGN KEY QRZ3_BLOB_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ3_CRON_TRIGGERS DROP FOREIGN KEY QRZ3_CRON_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ3_SIMPLE_TRIGGERS DROP FOREIGN KEY QRZ3_SIMPLE_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ3_SIMPROP_TRIGGERS DROP FOREIGN KEY QRZ3_SIMPROP_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ3_TRIGGERS DROP FOREIGN KEY QRZ3_TRIGGER_TO_JOBS_FK;
ALTER TABLE IGAQRZ.QRZ4_BLOB_TRIGGERS DROP FOREIGN KEY QRZ4_BLOB_TRIG_TO_TRIG_FK;
```

```
ALTER TABLE IGAQRZ.QRZ4_CRON_TRIGGERS DROP FOREIGN KEY QRZ4_CRON_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ4_SIMPLE_TRIGGERS DROP FOREIGN KEY QRZ4_SIMPLE_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ4_SIMPROP_TRIGGERS DROP FOREIGN KEY QRZ4_SIMPROP_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ4_TRIGGERS DROP FOREIGN KEY QRZ4_TRIGGER_TO_JOBS_FK;
ALTER TABLE IGAQRZ.QRZ5_BLOB_TRIGGERS DROP FOREIGN KEY QRZ5_BLOB_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ5_CRON_TRIGGERS DROP FOREIGN KEY QRZ5_CRON_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ5_SIMPLE_TRIGGERS DROP FOREIGN KEY QRZ5_SIMPLE_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ5_SIMPROP_TRIGGERS DROP FOREIGN KEY QRZ5_SIMPROP_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ5_TRIGGERS DROP FOREIGN KEY QRZ5_TRIGGER_TO_JOBS_FK;
ALTER TABLE IGAQRZ.QRZ6_BLOB_TRIGGERS DROP FOREIGN KEY QRZ6_BLOB_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ6_CRON_TRIGGERS DROP FOREIGN KEY QRZ6_CRON_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ6_SIMPLE_TRIGGERS DROP FOREIGN KEY QRZ6_SIMPLE_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ6_SIMPROP_TRIGGERS DROP FOREIGN KEY QRZ6_SIMPROP_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ6_TRIGGERS DROP FOREIGN KEY QRZ6_TRIGGER_TO_JOBS_FK;
ALTER TABLE IGAQRZ.QRZ7_BLOB_TRIGGERS DROP FOREIGN KEY QRZ7_BLOB_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ7_CRON_TRIGGERS DROP FOREIGN KEY QRZ7_CRON_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ7_SIMPLE_TRIGGERS DROP FOREIGN KEY QRZ7_SIMPLE_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ7_SIMPROP_TRIGGERS DROP FOREIGN KEY QRZ7_SIMPROP_TRIG_TO_TRIG_FK;
ALTER TABLE IGAQRZ.QRZ7_TRIGGERS DROP FOREIGN KEY QRZ7_TRIGGER_TO_JOBS_FK;
```

Performance is not affected by dropping the constraints, but the deadlocks disappeared from the hierarchy tests.

26. General Tips

Listed here are a few items which do not require much explanation, and thus do not warrant an entire section. These are general guidelines that would apply in most environments.

1. When possible, run Reports in the off-hours. With global workforces, global deployments, and round-the-clock VA usage, it might be difficult to find a time when the VA is at rest. However, there are very likely times when the VA will be less busy. Use this time to run reports, NightShift activities, campaigns, etc.
2. Configure smaller certification campaigns, hierarchies, or role mining jobs. Schedule these smaller jobs to run in periods of lower utilization.
3. Use the procedure listed in Collecting Java Core Dumps to periodically clear the system logs.

4. It is very important for the VA and the database tier to have synchronized clocks. A difference between the clocks can cause delays in operations and affect the accuracy of log comparisons during problem determination.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features contained in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM might have patents or pending patent applications that cover subject matter described in this document. The furnishing of this document does not grant you any license to these patents. Send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and

do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it to enable: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information might be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments might vary significantly. Some measurements might have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements might have been estimated through extrapolation. Actual results might vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources.

IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products.

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding the future direction or intent of IBM are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute

these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs that conform to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp.

2004, 2013. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations might not appear.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both: http://www.ibm.com/legal/copytrade.shtml

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names might be trademarks or service marks of others.