

## Provision a Blue Prism database server

This guide provides guidance for the provisioning of a database server to host a Blue Prism database.

The instructions and recommendations contained within this document should be considered as a guide only. There are a large number of options available when provisioning databases that may relate not only to database specific settings, but also the host instance, as well as all instances on a given server.

It is recommended that both industry standard best practice; and recommendations from an experienced database administrator are sought. These should be reviewed in light of the wider impact that they may have on the overall environment.

### Blue Prism databases

The Blue Prism database is a central repository which holds process definitions and audit information as well as configuration data such as environment-wide system settings. The database is specific to a Blue Prism environment and therefore within an organisation there may be a requirement to host a number of databases based on:

- The number of independent production environments
- Requirements for development, test, staging and pre-production environments.

From a co-existence perspective Blue Prism databases may reside on a single SQL instance or conversely may each be situated on an independent instance. Additionally, subject to capacity and performance considerations, Blue Prism databases may share a SQL instance with other application databases.

### Selecting a SQL Server/instance

When selecting the SQL Server or SQL Server instance to host the Blue Prism database(s) the following should be considered:

- Proximity of the SQL Server to the Blue Prism Application Server(s), and other Blue Prism resources, particularly when implemented across large or multi-site networks.
- The number of existing databases that share the underlying hardware (CPU, RAM etc.), the utilization of those databases, and the capacity available.
- If relevant, whether any existing SQL Server instances are already configured to offer high-availability or disaster recover capabilities (such as SQL Clustering, Replication, Mirroring etc.) which may be desirable for the production Blue Prism databases.
- Availability of disk space whilst also considering the level of resilience, performance and capacity for expansion (e.g. SAS disks versus SATA, RAID type, SAN based or direct attached storage etc.)

## Provisioning the database

Blue Prism database creation is typically product driven however options are available for a script to be provided allowing the database to be manually created. Once created it is advisable to review the configuration and settings for the Blue Prism database and SQL Server instance as detailed below.

### Database collation

The collation of the Blue Prism database must be one that is case-insensitive and that supports the 1252 codepage.

The case-insensitive feature is typically denoted by the presence of CI within the collation name e.g.:

- Latin1\_General\_CI\_AS
- SQL\_Latin1\_General\_CP1\_CI\_AS

### Snapshot isolation

From version 6.4.2, snapshot isolation must be enabled on all Blue Prism databases. To facilitate this, the *Allow Snapshot Isolation* setting is automatically enabled when configuring and upgrading Blue Prism environments to version 6.4.2 and above.

### Resource allocation

Microsoft SQL Server can be a resource-intensive application and the recommendations below help to maximise performance whilst also providing administrative control over how the resources will be allocated.

### Storage (disk space utilisation and configuration)

It is recommended that the following are installed on, or configured to, use independent drives to aid performance by reducing contention in read/write operations on the disks:

- **Server**
  - Server operating system.
  - SQL Server installation directory.
- **SQL Server Instance**
  - SQL Server tempdb.
  - SQL Server data files.
  - SQL Server log files.

For each high performance database it is recommended that the specific data files and logs should be configured to use independent dedicated drives.

Additionally consideration should be given to the location of the database backup directory to ensure that the backups will be retained in the event of the main database files becoming lost or corrupted.

## CPU and RAM allocation

The amount of processes and RAM that will be able for SQL processing is specified for each SQL Server instance. Often the default settings allow any one instance to fully consume all processors and RAM which can have a negative impact on the underlying operating system, and any other SQL Server instances on that server.

In order to prevent this behavior and manage the hardware capacity it is recommended that:

- Each SQL Server instance should have its settings reviewed to ensure that it is not able utilise more RAM than is appropriate.
- The databases within a given SQL instance are reviewed to ensure that they are not likely to require all of the resources allocated to that instance to the detriment of other databases.
- The total maximum RAM allocated across all SQL Server instances on a given Server does not exceed the total amount of RAM available (whilst also ensuring a suitable amount of RAM is available for the operating system). This helps to ensure each SQL instance effectively has a reserved amount of RAM that will always be available.

## Database growth

The expansion that occurs as the database data and logs increase in size can have an impact on database performance. This typically occurs on an ad-hoc basis as either the data or logs require additional space beyond that currently available within the space allocated to the existing files. The issue can be compounded when this expansion takes place in an uncontrolled manner and most seriously impacts the performance of database log files.

There are a number of settings that can be applied to the Blue Prism database(s) to reduce the occurrence of expansion and to provide greater controller to the database administrators.

### Set appropriate initial file sizes for data and logs

Setting the initial size for the data and log files to be sufficient for a defined period of time, such as the next 6 months, will reduce the need for such files to be expanded dynamically as part of query execution.

Suggested sizing for Blue Prism databases is contained within the appropriate Blue Prism infrastructure specification documentation and is typically based on a “per robot” basis allowing these initial sizes to be approximated in advance.

It is also recommended that a suitable initial size is also defined for the tempdb which is reset to its default size each time the SQL Server service is restarted. As this database services all other databases on a given SQL instance it is recommended to review the size that this database grows to on a periodic basis in order to establish a suitable initial value.

### Ensure auto-growth is allowed

Whilst it is recommended that database file and log growth is planned, the auto-growth setting should also remain turned on. This ensures that ad hoc growth is not prevented if required.

## Set appropriate growth limits

The amount by which the files will be expanded as auto-growth is required may be specified as fixed amounts, or as a percentage of the existing size of the file. In order to prevent growth being exponential, it is recommended that the auto-growth limits should be set to be a fixed size.

These fixed amounts should be reviewed to ensure that if auto-growth is required, that the size of the growth is appropriate to minimise the number of times auto-growth may be required in short succession. Considering that Blue Prism database growth and use typically correlates to the number of runtime resources, it may be prudent to set the auto growth values to be: 100 MB for the data file; and 50 MB for the log file.

## Enable Instant File Initialization

This feature typically benefits the speed at which transaction log files can be grown automatically by removing the step of zero-initialisation from the process.

## Turn off auto-shrink

The auto-shrink capability periodically reviews the data and log files for unused space and attempts to shrink the files in order to release the space. This processing in itself is resource intensive and also works against the premise of both: manually controlling the data and log file sizes; and reducing the need for dynamic growth of the database files. Additionally shrinking typically causes file fragmentation which in turn results in additional performance degradation.

## Backup

It is strongly recommended that for all Blue Prism databases that a regular backup of the database takes place to a secure, remote location to cater for a range of disaster recovery scenarios.

If the database has been set to use a Full recovery model, it is important that regular transaction log backups also take place. This not only allows for a point in time recovery to take place, but also provides free space within the log files to reduce the amount of on-going log growth required.

## Database statistics

Database statistics are used by Microsoft SQL Server to decide how a query should be executed and therefore it is imperative that these are always up-to-date and accurate. This can be helped by ensuring the following database options are turned ON for the Blue Prism database(s):

- AUTO\_CREATE\_STATISTICS
- AUTO\_UPDATE\_STATISTICS

## SQL permissions

The minimum SQL permissions required on the Blue Prism database for business as usual or normal operation are listed below:

- Datareader
- Datawriter
- [All roles prefixed with bpa\_] E.g.
- bpa\_ExecuteSP\_DataSource\_bpSystem
- bpa\_ExecuteSP\_DataSource\_custom
- bpa\_ExecuteSP\_System

The roles prefixed “bpa\_” are only available once the database has been configured using the in-product Create Database functions or manually using the CreateScript.

The minimum SQL permissions do not provide appropriate privileges to carry out Create, Configure or Upgrade database actions, therefore an appropriate administrator account will need to be used when any of these actions are required:

- Create database – dbcreator (server role) or sysadmin (server role)
- Configure database – sysadmin (server role) or dbowner (database role)
- Upgrade database – sysadmin (server role) or dbowner (database role)

To manually execute the Create or Upgrade database scripts (available via Blue Prism Support) against an existing database, the following SQL permissions are required by the user carrying out the actions:

- DBCreate: sysadmin (server role) or dbowner (database role)
- DBUpgrade: sysadmin (server role) or dbowner (database role)

## Virtualized database servers

The performance of the overall Blue Prism platform can be significantly affected by the performance of the Blue Prism database and therefore it is essential to ensure that it is appropriately resourced and remains highly responsive at all times. For production environments it is commonly not appropriate to virtualize this component.

If there is a desire to virtualize the component it is essential to assure appropriate levels of physical hardware resource are provided, and that the appropriate guidelines from both Microsoft and the virtualization technology provider (e.g. Citrix, Microsoft, VMWare etc.) are adhered to. SQL Server databases are extremely sensitive to the performance of the underlying hardware and therefore if the virtualized resources are not reflective of the physical resources (CPU, RAM, disks) that would otherwise have been provided, the performance of the Blue Prism environment will be affected. Consider also if the underlying resources are shared by additional components this will result in contention which will further impact performance.

The virtualization of components commonly coincides with a sharing of, or reduction in, the level of physical resource (CPU, RAM, disk space / performance) available which will significantly impact the performance the database.

Whilst not an exhaustive list, the most common challenges of virtualizing this component include:

- **CPU contention** – Commonly the number of virtual CPU cores (vCPU) available on a given virtualization host, significantly exceeds the number of physical CPU cores (E.g. 48 vCPUs may represent 16 physical CPU cores). The amount of over-allocation of the CPU cores must be considered because, in this example, if the SQL Server was allocated 4 vCPUs, this would be equivalent to the SQL Server having 1.3 physical CPU cores rather than 4 CPU cores as may be expected.
- **Disk performance** – When deploying a physical SQL server according to the vendor guidelines there will be a number of separate, dedicated, high performance disk arrays for each of a number of purposes including: operating system, SQL server installation files, SQL server data files, SQL server log files, SQL server backup files, SQL Server temp db.

Virtualized servers commonly share an underlying disk subsystem with other devices and whilst there may be separate logical drives, the performance of these disks is likely to be significantly reduced in comparison.

## Blue Prism database usage patterns

Communication between the Blue Prism runtime resources, application servers and database is typically moderate to high in volume, and transactional in nature as records are frequently inserted into the session log, along with look-ups and updates being performed within workflow tables.

Consideration should be given to the proximity of the database server to the Blue Prism application server and runtime resources, particularly when implemented across large or multi-site networks. Where network latency is an issue, it will be made more prominent by the frequency of the queries performed.

Commonly the Blue Prism database will receive direct connections only from each Blue Prism application server within a given environment.

In some circumstances, such as where application servers are not deployed, any Blue Prism component can be configured to establish a direct database connection. This will be subject to the application of appropriate routing, authorization and access settings.

The number of connections that will be established by each directly connecting device is managed by the .NET Framework through use of SQL connection pools.

# Maintain a Blue Prism database server

The instructions and recommendations contained within this section should be considered as a guide only. There are a large number of options available when maintaining databases that may relate not only to database specific settings, but also the host instance, as well as all instances on a given server.

It is recommended that both industry standard best practice and recommendations from an experienced database administrator are sought. These should be reviewed in light of the wider impact that they may have on the overall environment.

## SQL Server Maintenance

### Database server settings

There are a number of SQL Server settings that should be reviewed to promote the performance and stability of the SQL Server platform. These settings vary in their scope and may affect: the Blue Prism database; the specific SQL instance that hosts it; and all SQL instances on a given server.

It is strongly recommended that the respective Blue Prism documentation is reviewed in relation to the settings as they are critical in promoting a stable and reliable SQL platform.

The settings to be considered relate:

- Resource (RAM and CPU) allocation
- Storage utilisation and configuration
- Auto-growth
- Database statistics
- Use of Auto-Shrink



## Shrink operations

Shrink commands, which may be featured as part of SQL maintenance plans, should be avoided because they cause significant amounts of fragmentation and are often unnecessary as databases re-use the released space as they re-grow. Additionally performance is impacted less during growth if the space has already been allocated and is available (rather than requiring dynamic expansion).

It is recommended that activities which include shrinking either the database, data files, or log files, are avoided. There are a limited number of scenarios where it is appropriate to use a shrink operation, however shrinking typically results in performance degradation.

## Backup

The Blue Prism databases must be backed up regularly and it is recommended that the backup storage location is a secure, remote location to cater for a range of disaster recovery scenarios.

The backup model should be appropriate to the criticality of the Blue Prism solution (which may vary across the environments) e.g. it is likely that the production environment will require a higher level of resilience and recovery than the development environment).

Blue Prism supports both **Simple** and **Full** SQL recovery modes and it is recommended that the benefits of each is reviewed to ensure the method chosen is appropriate to the criticality of the solution.

If the database has been set to use a **Full** recovery model, it is important that regular transaction log backups take place to periodically provide free space within the log files and restrict the amount of on-going log growth required.

## General maintenance

The following maintenance tasks should be carried out periodically – these can typically be achieved using standard IT tools:

- **Available physical disk space** – The amount of physical disk space available on each disk array or partition used by the SQL Server should be reviewed to ensure appropriate free space is available. This is especially important where a disk partition is shared by multiple databases or applications.
- **Disk array fragmentation** – The amount of disk fragmentation should be reviewed to establish if a defragmentation task is required. Fragmentation causes performance degradation in read/write operations to the disk which in turn will impact the general performance of the SQL server. This is of particular importance for the disk array on which the SQL log files reside.
- **Paging file** – The paging file is used by the operating system as part of the management of RAM and therefore it should be set to an appropriate size
- **Event viewer** – The event viewer on the SQL Server should be reviewed for errors as these could be indicative of underlying issues.
- **Verify maintenance plan and backup execution** – It is important to make sure that the scheduled maintenance and backup tasks have taken place as expected, and without error. Database backups should also be verified to ensure that they can actually be restored if required. Options such as RESTORE VERIFY ONLY and RESTORE can be considered for this type of verification although both are intensive processes so may be best applied within a non-production environment.

## Database maintenance

In order to maximise the performance of the database it is important that regular maintenance takes place. Maintenance tasks can also be used to identify issues that may be occurring at the database level before they have become apparent.

The following maintenance activities should be undertaken regularly, ideally on a scheduled basis.

### Verify data integrity

Use of DBCC CHECKDB will check the logical and physical integrity of all of the objects in the specified database. If any issues are found there are a number of repair procedures available, however it should be noted that the resolution for some issues may require the database to be overwritten with a restore of a previously taken backup.

### Identify and resolve index fragmentation

The performance of the indexes is important to the overall performance of the database and therefore it is recommended that the level of fragmentation is reviewed and appropriate steps taken to defragment the indexes as required.

Whilst there are few out of the box features that provide insight into the level of index fragmentation, online resources contain examples of scripts that can help with checking the level of fragmentation across all indexes in a database.

Commands such as ALTER INDEX can be used to rebuild or re-organise the indexes.

Consider also the fragmentation of the underlying file system as this may further impact the performance of the SQL Server.

### Ensure statistics are up-to-date

Database Statistics are used by the Query Processor in Microsoft SQL Server to determine how a query should be executed. It is therefore imperative that these are always up-to-date and accurate.

The most common way to generally ensure that statistics are maintained is to turn on the following database settings:

- Auto Create Statistics
- Auto Update Statistics

When updating statistics manually it is important that the action includes carrying out a full scan – this can be ensured by using the alter index action to carry out an index rebuild as this uses a full scan and updates the index statistics automatically. This action however does not update column statistics – this can be done using the UPDATE STATISTICS WITH FULLSCAN, COLUMNS statement.

### Free space within SQL Data/Log files

The amount of free space available within the data and log files should be regularly evaluated to establish whether the size of the data or log files needs to be increased to provide additional free space. The act of expanding the files dynamically at the time-of-need can have a performance impact, particularly when log files grow, so it is recommended that where possible, growth should take place manually and at a predetermined time.

## Evaluating file size and % free space

### Log Files

The command `DBCC SQLPERF(logspace)` can be executed against the master database to provide the required information for all log files in a given SQL instance.

### Data and Log Files

The method `FILEPROPERTY` can be used as part of a query executed against the master database to provide the required information for all data and log files in a given SQL instance.

```
SELECT DB_NAME() AS DatabaseName,  
name AS FileName,  
size/128.0 AS FileSize_MB,  
size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT)/128.0 AS FreeSpace_MB  
FROM sys.database_files;
```

Whilst databases should be allowed to grow automatically if they need to, the ideal scenario is that the file sizes are increased manually in advance ready for use by SQL when the space is required.

## Periodic review of the SQL database settings

The SQL Database Settings referenced within this document should be reviewed periodically to ensure that no settings have inadvertently been changed, and also as an opportunity to ensure that any additional SQL databases or instances have not been configured with settings that may conflict with the instance hosting the Blue Prism database(s).

## Blue Prism in-product maintenance

The Blue Prism software includes a number of maintenance capabilities which are predominately focused on removing old information from the database to prevent ongoing database growth and expansion.

### Archiving

Blue Prism Archiving is used to extract specific data types of a certain age from the database for the purposes of retaining historic data externally for prolonged periods of time without requiring ever-increasing amounts of space within the SQL database.

The facility to re-import archived data is provided which allows the Blue Prism GUI to be used to view and traverse the historic information.

The following data is currently affected by archiving:

- Session logs are extracted into a compressed XML file and removed from the database.
- Process alerts that relate to the sessions logs being extracted are permanently deleted.

The process of archiving relies on the SQL database and therefore manual archiving should be performed under controlled conditions to limit the impact of this action on other users. Additionally the SQL actions performed by archiving will cause SQL transaction logs to be generated and therefore it is recommended that, if the database is set to Full Recovery Mode, that a log backup is taken shortly after large archiving operations have completed to reduce the log size within the file.

Blue Prism recommends that a centralised archive location is defined and configured for appropriate access by the relevant users and resources that will be responsible for archiving. The permissions granted for the location will need to include the ability to create folders, and read/write files.

There are three options for archiving:

- Manually via the user interface.
- Automatically via the user interface.
- Using the command line (automatec.exe).

The automatic and command line archiving actions provide the advanced option to permanently delete the applicable data without first extracting the data to folder location.

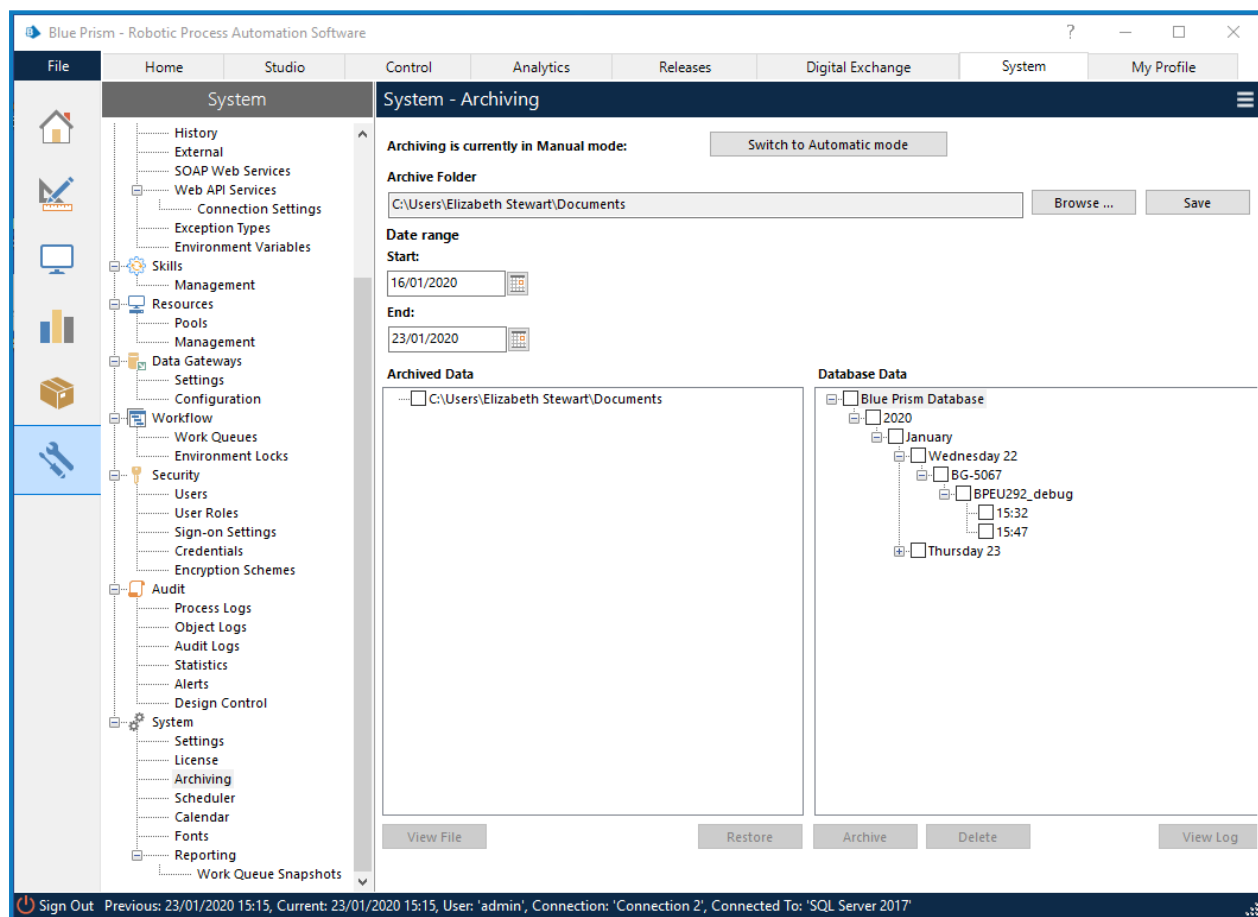
## Manual initiation

Within Blue Prism the logs are displayed using a tree structure based on the:

- year, month and day that a given process was started.
- name of the process.
- runtime resource which executed it.
- time at which it started.

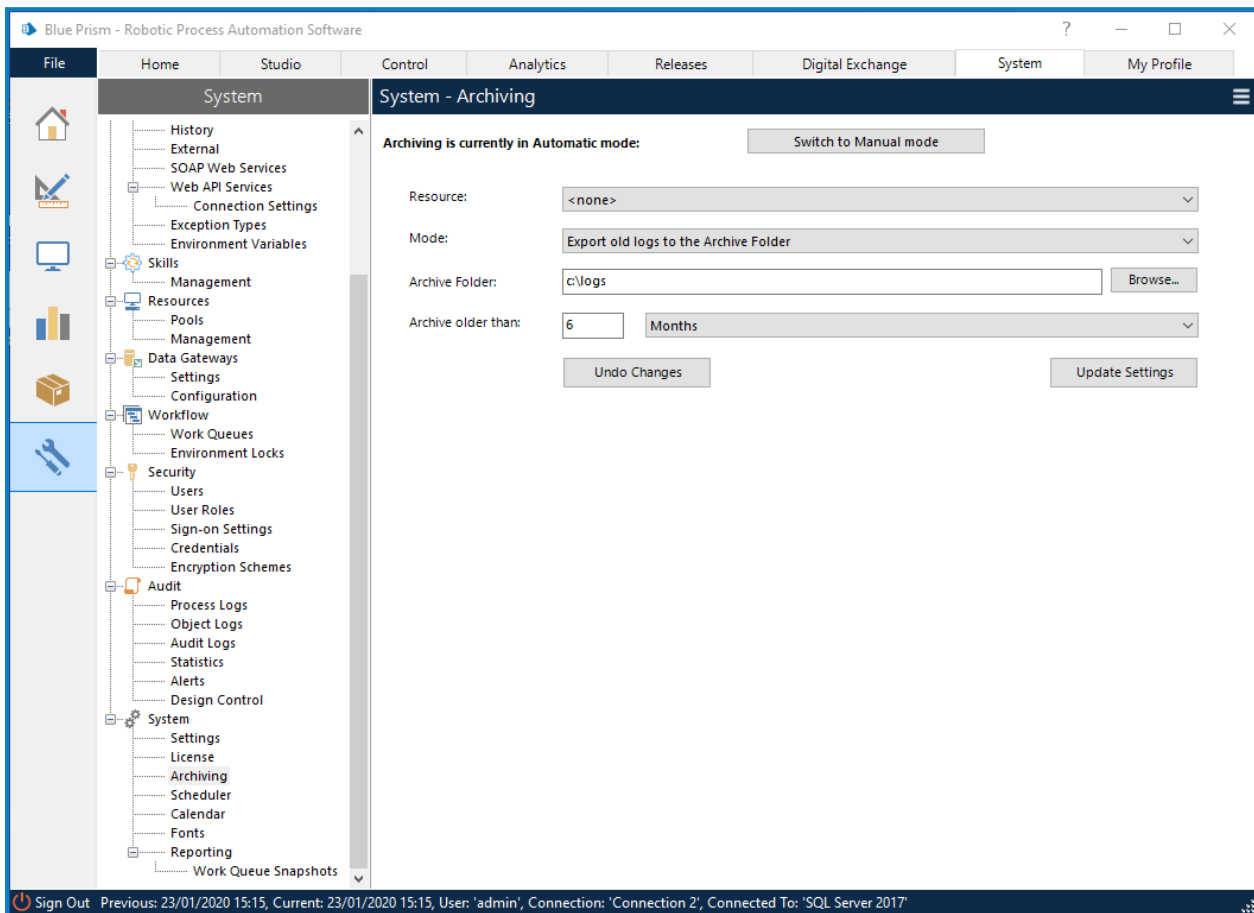
Manual archiving is initiated by navigating to the **Archiving** tab (System Manager -> System -> Archiving) and ensuring that **manual** mode is selected.

The user can select a number of individual items or branches in the tree for export, as well as the destination folder where the archived information will be stored. Blue Prism creates a folder structure within the target location representing a similar structure to how the data is presented via the Blue Prism user interface.



## Automatic, scheduled initiation

Scheduled archiving is configured by navigating to the **Archiving** tab (System Manager -> System -> Archiving) and ensuring that **automatic** mode is selected.



With automatic mode there are a number of configurable options:

- **Resource** – The name of the resource that will be used for the archiving operation.
- **Mode** – Provides the option to either: export and archive the files prior to automatically removing the data from the system; or alternatively to permanently delete the data without archiving.
- **Archive folder** – The location where the archived data will be placed. Blue Prism creates a folder structure within the target location representing a similar structure to how the data is presented via the Blue Prism user interface.
- **Archive older than** – A configurable value that indicates what age the data must be to be processed by the scheduled archive action.

The frequency at which archiving procedures are initiated is automatically determined by Blue Prism to encourage frequent, low-volume, archiving rather than occasional processing of larger amounts of data.

## Command line initiation

A command line method of AutomateC.exe can be used to trigger an archive. Archived files exported using the command line will be placed in the folder location specified on the **Archiving** tab within Blue Prism **System Manager**.

If configured for **single sign-on** the command to archive data older than 3 months is:

```
AutomateC.exe /archive /sso /age 3m
```

If configured for **native** authentication the same command is:

```
AutomateC.exe /archive /user "username" "password" /age 3m
```

When triggered by the command line the archiving process is carried out by the resource on which the command is executed. It is therefore necessary for the chosen resource to have full read/write/execute access to the folder location specified within Blue Prism.

If there is a desire to simply **delete the archived information rather than storing it to a folder location**, the /delete switch can be used:

```
>AutomateC.exe /archive /sso /age 3m /delete  
AutomateC.exe /archive /user "username" "password" /age 3m /delete
```

Use of the /delete switch permanently deletes the data and therefore removes the ability to retain or re-import the archived information at a later time.

## Audit logs

There is currently no in-product mechanism for deleting audit logs. The tables are generally quite small and grow slowly, except for the inclusion of historic versions of process and business objects.

## Database updates

Periodically Blue Prism software updates also require updates or changes to the Blue Prism database. Such database updates can be manually driven from within the product, or alternatively SQL scripts can be provided to allow the updates to be applied manually on the SQL Server.

It is recommended that a full database backup takes place and is subsequently verified prior to any manual database updates or modifications.

## Session logs

There are a number of options for managing and deleting session log information:

- **Using the Archiving feature (recommended)** – This includes manual, automatic and command line archiving (or permanent deletion) of session logs and can be aligned with the organisation's data retention policy.
- **Via the User Interface** – Within Control Room, select the items to be deleted and choose to Delete.

## Work queues

Historic work items can be permanently deleted using any of the following methods:

- **Via the User interface** – Within Control Room, select the items to be deleted and choose to Delete Worked Cases.
- **Using a Blue Prism process** – A custom Blue Prism process can be configured to review and delete the appropriate items using the Delete Processed Items action.
- **Command line** – AutomateC.exe includes a /queueclearworked command which can be used to permanently delete worked items.

It is important to ensure that work items are not required as part of a data retention policy prior to deleting them as there is presently no facility for archiving or re-importing them.