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, for example:

- Latin1_General_CI_AS
- SQL_Latin1_General_CP1_CI_AS
- Chinese_PRC_CI_AS
- Japanese_CI_AS
- Arabic_CI_AS
- Greek_CI_AS
- Cyrillic_General_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 utilization 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 or Upgrade database actions from within the product, therefore an appropriate administrator account will need to be used when any of these actions are required:

- Create database – sysadmin (server role)
- Upgrade database:
  - When deleting the existing database – sysadmin (server role)
  - When not deleting the existing 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)
- DBUpgrade: sysadmin (server role) or dbowner (database role)
  - When deleting the existing database – sysadmin (server role)
  - When not deleting the existing database – sysadmin (server role) or dbowner (database role)

For more information, see Minimum SQL permissions.

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.