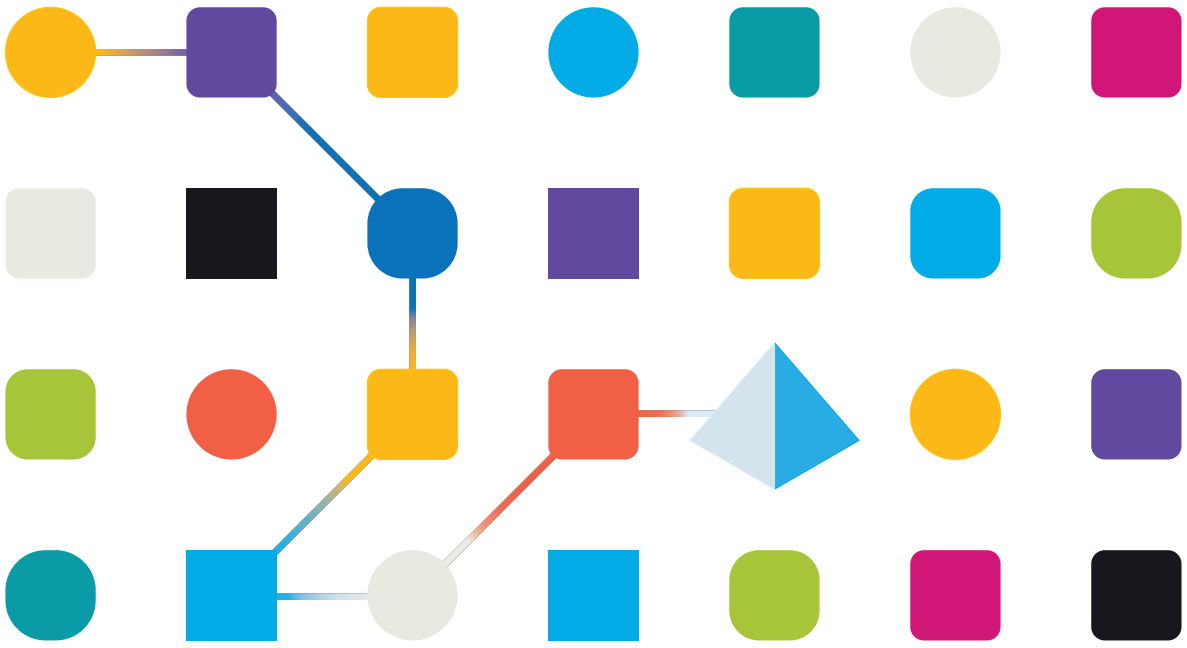
Blue Prism 7.0

Data Gateways

Document Revision: 3.0



Trademarks and Copyright

The information contained in this document is the proprietary and confidential information of Blue Prism Limited and should not be disclosed to a third-party without the written consent of an authorized Blue Prism representative. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying without the written permission of Blue Prism Limited.

© 2024 Blue Prism Limited

"Blue Prism", the "Blue Prism" logo and Prism device are either trademarks or registered trademarks of Blue Prism Limited and its affiliates. All Rights Reserved.

All trademarks are hereby acknowledged and are used to the benefit of their respective owners. Blue Prism is not responsible for the content of external websites referenced by this document.

Blue Prism Limited, 2 Cinnamon Park, Crab Lane, Warrington, WA2 0XP, United Kingdom.
Registered in England: Reg. No. 4260035. Tel: +44 370 879 3000. Web: www.blueprism.com

Contents

Introduction 1
Configure Data Gateways for Windows authentication 3
Prerequisites 3
Authentication options for SQL login 3
Install the Data Gateways engine components 3
Create a Data Gateways login using a Windows authentication user 3
Enable the Data Gateways process 4
Configure Data Gateways settings 7
Create a Data Gateways configuration 9
Configure Data Gateways for SQL authentication 12
Prerequisites 12
Authentication options for SQL login 12
Install the Data Gateways engine components 12
Create a Data Gateways login using a SQL authentication user 13
Enable the Data Gateways process 14
Create SQL Data Gateways credential in Blue Prism 16
Configure Data Gateways settings 19
Create a Data Gateways configuration 21
Data Gateways output types 24
File 24
HTTP endpoint 25
Splunk 25
Database 26
Data Gateways advanced outputs 29
Manage a Data Gateways configuration 30
Copy an output 30
Delete an output 30
Edit an output 30
Delete selected output 30
Data Gateways custom configurations 31
Create a custom configuration 31
Manage Data Gateways processes in Control Room 33
Start and stop the Data Gateways engine 33
Data Gateways VBO 34
Data Gateways user role permissions 35
Data Gateways configuration files 36
Input 36
Filter 37
Output 37
Event structure 37
Directing data to outputs based on content 40

Advanced configuration for database outputs 41
Custom configuration examples 41
Credentials in custom configurations 43
Data Storage when endpoint unavailable 44
Temporary data storage 44
'Dead letter queue' data storage 44
Blue Prism output extensions 44
File and custom output types 44
Troubleshooting Data Gateways 45

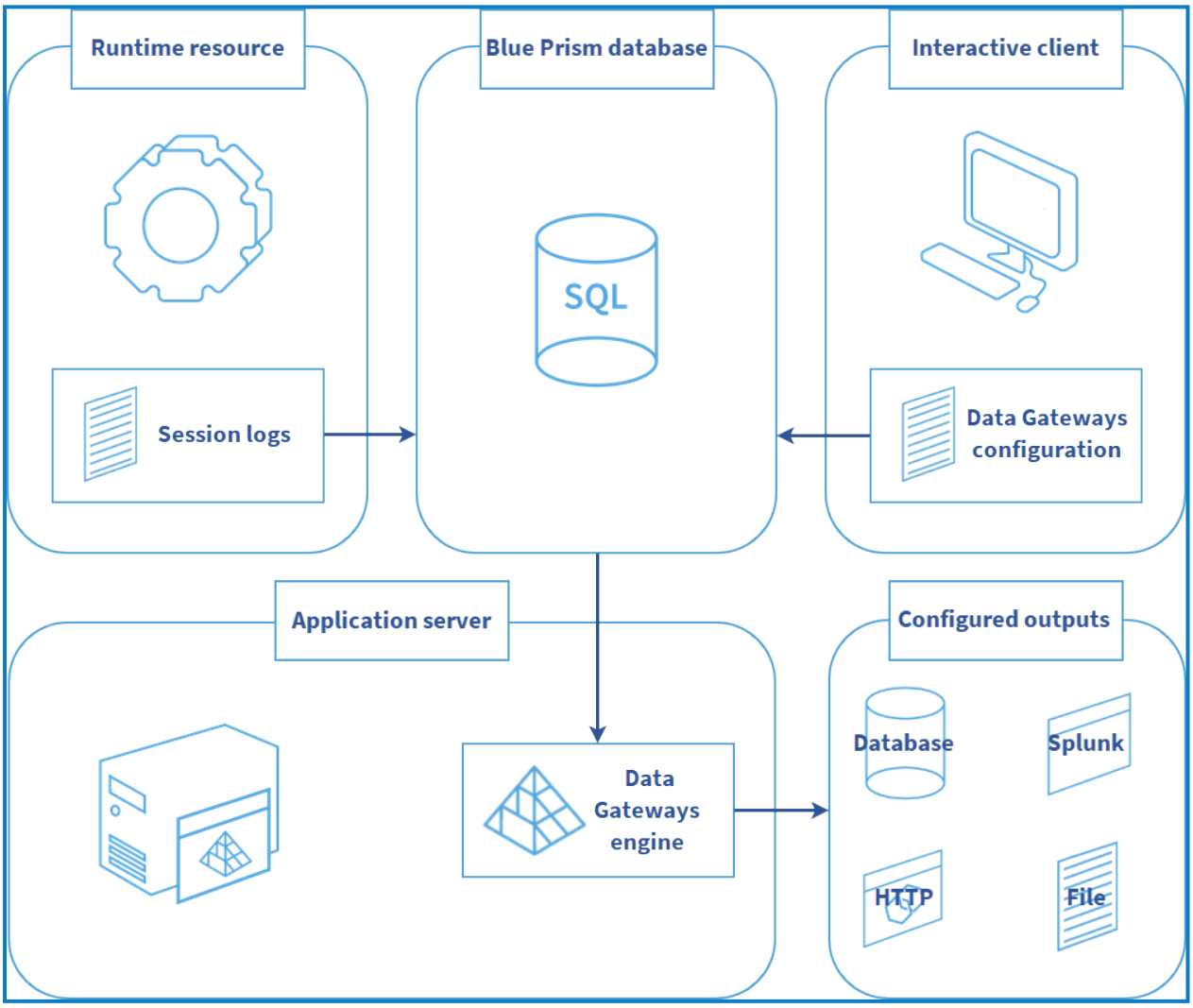
Introduction

Data Gateways provides an easy-to-use, centralized method to push data out of Blue Prism for use in external systems for monitoring and reporting, long-term data storage, and to feed machine learning models. Advanced configuration methods allow data to be directed to any required target. The data can be visualized and analyzed to provide valuable insight about Blue Prism environments without having to manually build similar capabilities into each relevant process automation.

By providing the ability to store data outside the Blue Prism database, organizations can use Data Gateways to support flexible data storage requirements. For example, an organization might want to save all their session data outside the Blue Prism database, or they might choose to store data in the database for a shorter period of time, pushing out a copy of that data for longer term data storage.

Settings are applied to determine what data will be processed by the Data Gateways engine, and a configuration defines the outputs to which data will be pushed. Data from session logs, published dashboards, process stages, and work queue analysis can be sent to a variety of external outputs - HTTP endpoints, external databases, third-party analysis tools, and flat files - providing flexibility and control over data analytics and storage.

The Data Gateways engine utilizes a number of Logstash components to send session log data to configured outputs.



Blue Prism collects the data to be sent to Data Gateways according to the configured Data Gateways settings. The data is stored centrally in a Blue Prism database. The Data Gateways engine then processes that data based on the configured rules and sends the data onwards to the configured outputs.

An appropriate Data Gateways engine must be installed to use Data Gateways with the Blue Prism versions listed below. It is advised to use the latest version available. To upgrade to a newer version of Data Gateways, first uninstall the existing version and then install the new version. The configuration data from the previous version of Data Gateways will be retained in the new version.

Blue Prism	Data Gateways engine
7.0	1.3, 1.4
6.10	1.3, 1.4
6.9	1.2, 1.4
6.8	1.2, 1.4
6.7	1.1, 1.4
6.6	1.1, 1.4
6.5	1.0, 1.4

Configure Data Gateways for Windows authentication

Prerequisites

The following conditions must be met before configuring Data Gateways:

- The Blue Prism version must be version 6.5 Enterprise Edition or above, which must be configured to connect via at least one application server.
- Data Gateways requires a dedicated SQL server login with access to the Blue Prism database, created in SQL server. Whether using SQL or Windows authentication a user must exist with access to the Blue Prism database using the BPA_DataGateways role.
- The user account that runs the Blue Prism application server process must have read and write permissions to the file directory where the Logstash configuration file (bpconf.config) is stored. By default this is found in C:\Program Files\Blue Prism Limited\Blue Prism Automate.
- Data Gateways must be run as a single instance on an application server. Multiple instances cannot run on the same application server at the same time.

  If failover capacity is required in the case of the application server becoming unavailable, a secondary Data Gateways instance can be set up on a separate application server in the environment and manually started. After the new Data Gateways instance has started, the initial instance will disconnect and the most recently started instance will take over.

Authentication options for SQL login

You need to create a SQL login to access the Blue Prism Data Gateways database (this is described in the relevant authentication method below). You can then use one of the following authentication methods:

- Windows authentication (integrated security), described in this section, using either existing or dedicated (recommended) credentials. If using existing Windows authentication credentials, the Data Gateways process will run in the same user context as the Blue Prism application server service.
- SQL authentication using credentials configured via Blue Prism Data Gateways configuration. The service will run under the context of the Data Gateways service. See Configure Data Gateways for SQL authentication for details if this is your chosen authentication method.

Install the Data Gateways engine components

Data Gateways requires a number of components to be installed and configured in the environment in which application servers are running the Data Gateways processes. The Data Gateways components are included in a separate installer, *Blue Prism Data Gateways Components*, available from the Blue Prism portal, see https://portal.blueprism.com/product/extras.

Create a Data Gateways login using a Windows authentication user

Data Gateways requires a SQL login, and a Windows authentication or SQL user to be created against the Blue Prism database. This section describes how to create a SQL login and a Windows authentication user.

Use Microsoft SQL Server Management Studio to create a SQL login using Windows authentication and to grant it the BPA_DataGatewaysEngine role on the Blue Prism database. Alternatively, run the SQL script below.

Create SQL login

To create a SQL login, run the following command:

```
CREATE LOGIN [loginname] FROM WINDOWS;
```

Before running the query, replace *loginname* with the SQL login credentials name. For example:

```
CREATE LOGIN [Data Gateways SQL Login] FROM WINDOWS;
```

Enable the Data Gateways process

This section describes how to enable Data Gateways in BP Server and applies to SQL authentication and Windows authentication users.

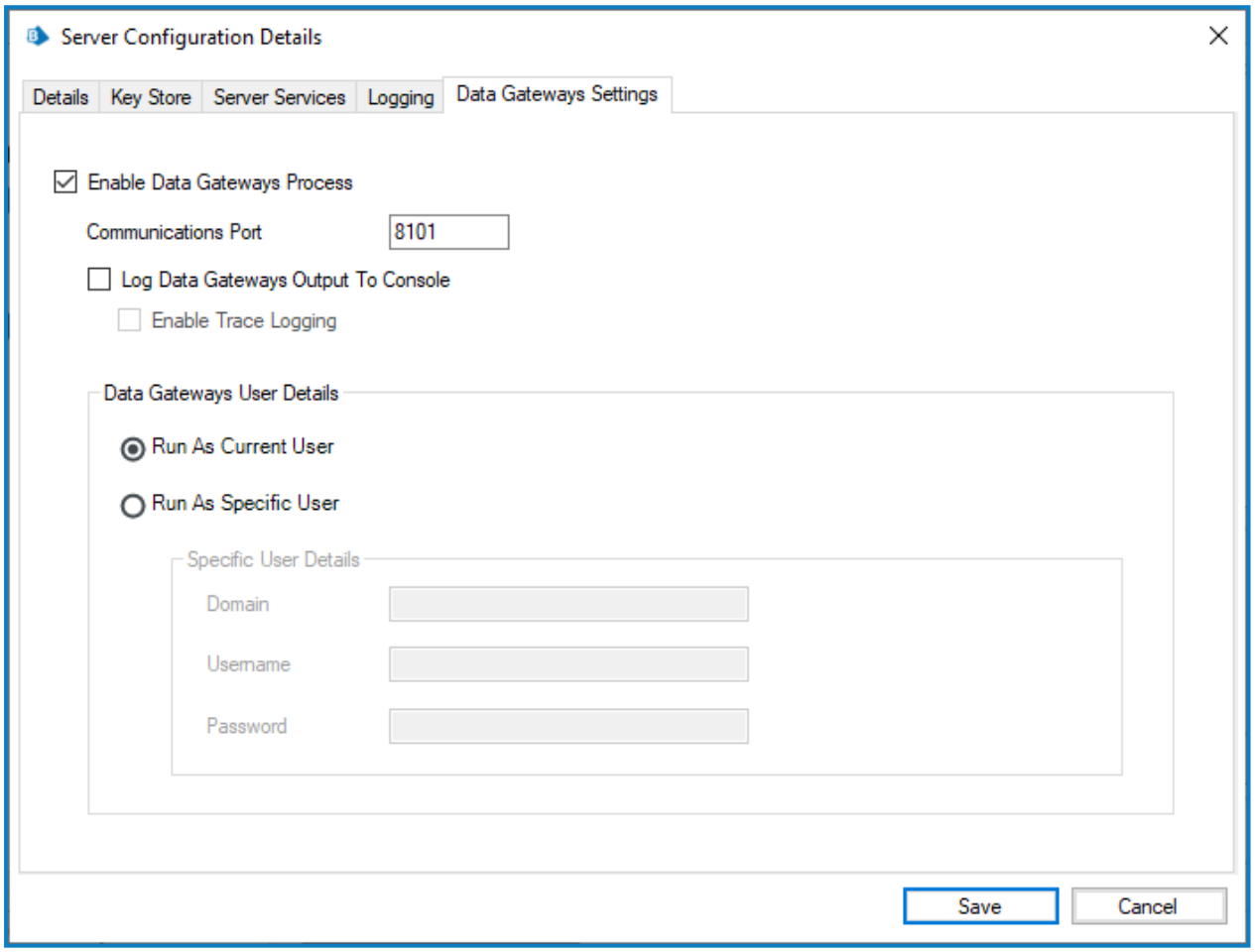
The Data Gateways process must be enabled and the associated port defined on all application servers that are required to run Data Gateways.

1. Open BPServer.exe and open the server configuration for the relevant environment.
2. Click the **Data Gateways Settings** tab.

3. Select **Enable Data Gateways Process** and enter the port that the application server will listen on for commands to start or stop the Data Gateways process.

   This port is also used by other Blue Prism application servers within the same environment to control the Data Gateways engine. This can be left as the default port (8101) unless there is another process on the application server that is already using that port. Ensure that security systems, such as firewalls, do not prevent other application servers from communicating with the Data Gateways engine via this port.

   The application server must be stopped before changing the port number.



4. If required, select **Log Data Gateways Output To Console** to send high-level Data Gateways messages to the console log in BPServer. This data can be used to help diagnose any potential issues with the Data Gateways integration. This option should not be left selected when not specifically required.
5. Select **Enable Trace Logging** to enable verbose Data Gateways logging, which can be used to further diagnose any issues between Blue Prism and Data Gateways.

6. In Data Gateways User Details, select one of the following options:
   - **Run as Current User** – The Data Gateways engine will operate in the same context as the Blue Prism Server service. Select this option if the Data Gateways engine will log into the Blue Prism database using a SQL authentication account. The credentials to use will be specified separately via the Data Gateways Configuration.
   - **Run as Specific User** – The Data Gateways engine will operate under the context of the Windows or Network account provided. Select this option if the Data Gateways engine will log into the Blue Prism database using a Windows Authentication account. It will access SQL using the specified credentials. If you select this option, additional configuration is required on the application server running the Data Gateways engine to configure a temporary folder that the Logstash components running under this context can access. See Additional configuration required to 'Run as Specific User for details.

It is not recommended to use *Run as Current User* to access the Blue Prism Database using Windows authentication. This configuration will result in the database being accessed using the account that is running the Blue Prism Server service which could result in the Data Gateways functionality having greater access to the database than it should have.

If using integrated security, we recommend using your own certificate to encrypt the Blue Prism server configuration files to make the configuration data more secure. This can be configured via *Manage Config Encryption Settings* in BPServer.exe.

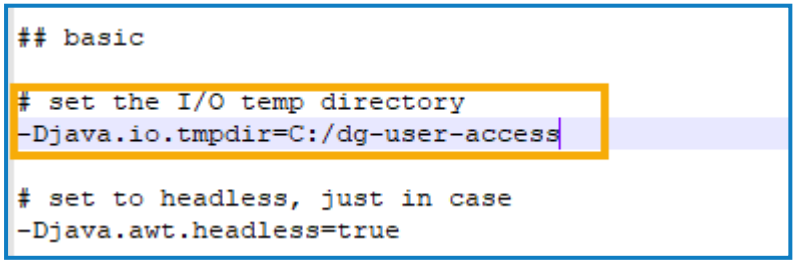
9. Click **Save** to apply the settings.
10. Start the Blue Prism Server Service from the Windows Services menu.

Additional configuration required to 'Run as Specific User'

If you have configured Data Gateways to *Run as a Specific User* you will need to perform the following steps to configure a temporary folder that the Logstash components running under this context can access. Typically *Run as a Specific User* is selected when you require the Data Gateways engine to authenticate against the database using Windows authentication.

The following steps must be performed on the application server running the Data Gateways engine:

1. Create a new temporary directory on the C:\ drive, for example *dg-user-access*.
2. Assign the default *Authenticated Users* permissions to the directory.
3. Open the *C:\Logstash\logstash\config\jvm.options* file in a text editor.
4. Add the temporary directory's path to the `-Djava.io.tmpdir=` line. Uncomment the line if it is commented out (remove the # from the beginning of the line).

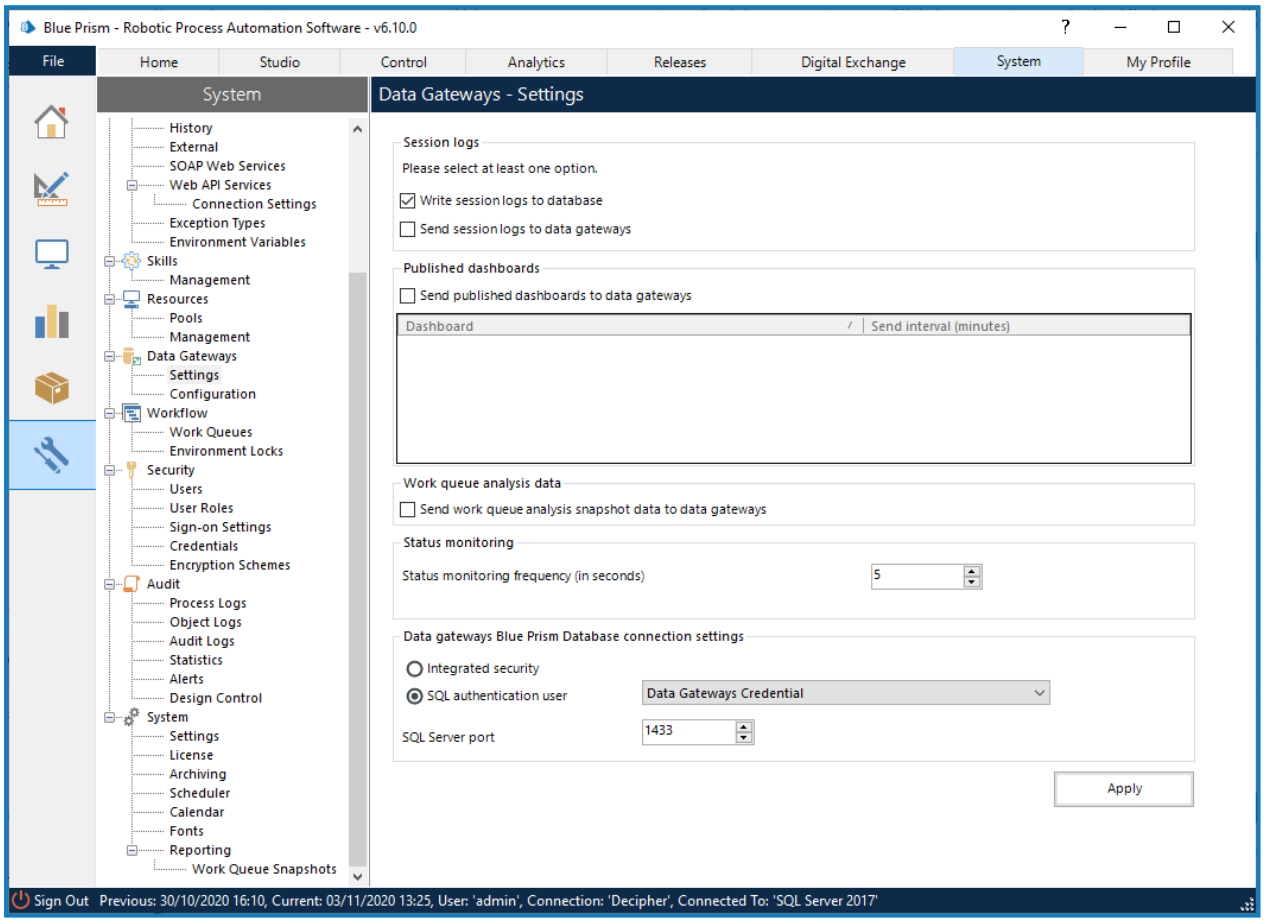


5. Save the changes.
6. Restart the Data Gateways service or the application server to apply the updates.

Configure Data Gateways settings

Configure the Data Gateways engine by determining what data will be processed and where relevant, the frequency at which data is sent.

1. From the **System** tab, select **Data Gateways** > **Settings**.



2. Select the required options to determine where session log data will be stored - at least one option must be selected:
   - **Write session logs to database** – Session logs will be sent to the session log table in the Blue Prism database. If not selected, the functionality to view new session logs in the Blue Prism client will be unavailable. This is the default setting and should only be changed if you no longer want new session log records to be accessible via the Blue Prism user interface.
   - **Send session logs to data gateways** – Session logs will be sent to a temporary storage table on the Blue Prism database where they are accessed by the Data Gateways engine for use in the configured outputs. The logs will be deleted from the temporary table after they have been processed by Data Gateways.

     If the specified HTTP, Splunk, or Database Data Gateway endpoint is not reachable when the data is being processed, the data is stored temporarily until the endpoint becomes available. See Data Storage when endpoint unavailable for details.

3. Select **Send published dashboards to data gateways** to send data from published dashboards to a database table in the Data Gateways system and set the frequency that data is sent for each dashboard. For more information about Blue Prism dashboards, see Dashboards.

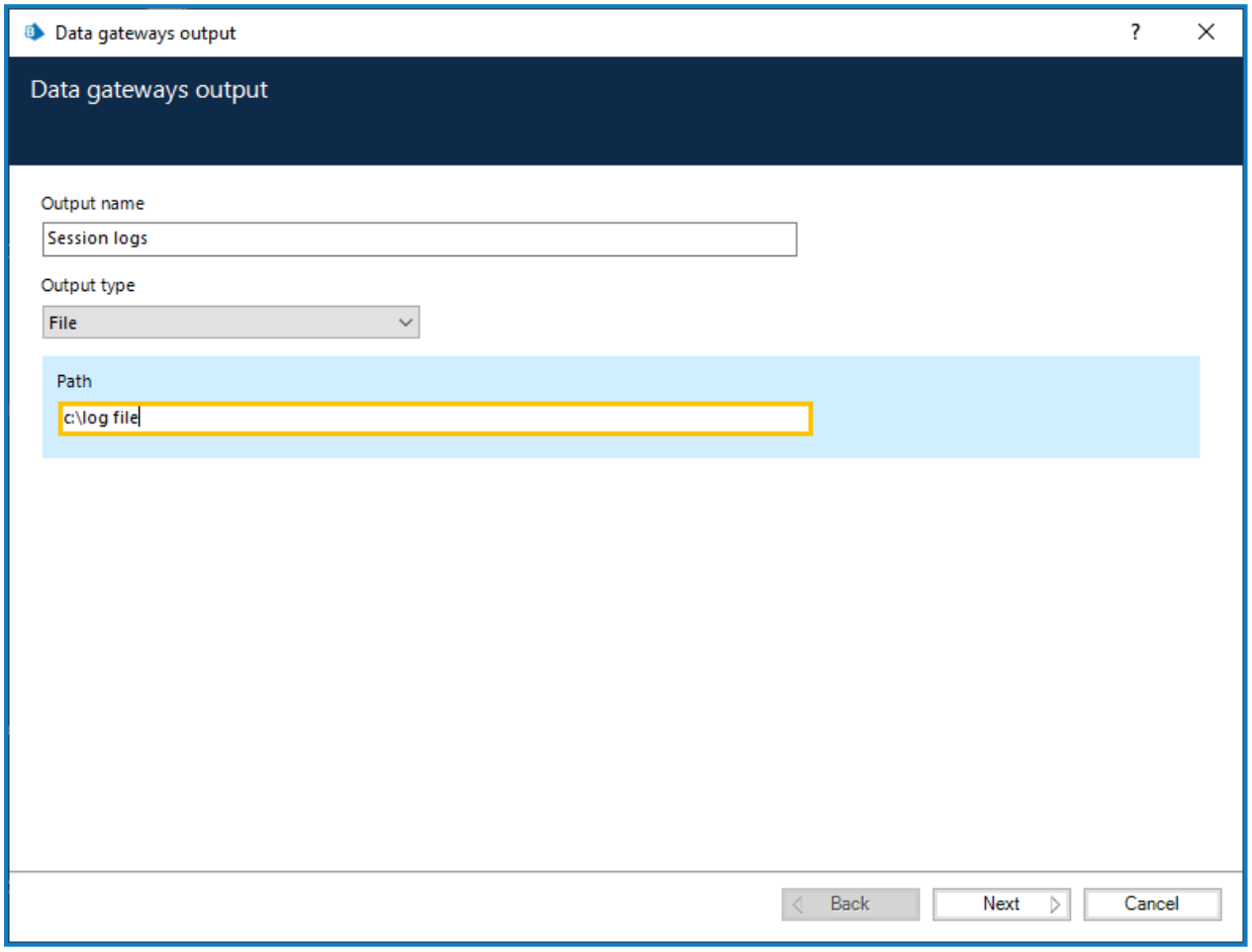
4. Select **Send work queue analysis snapshot data to data gateways** to send work queue analysis data to the database. For more information about Blue Prism work queue analysis snapshots, see the Work queue snapshots topic in the in-product help.
5. Set the **Status monitoring frequency** to a value between 5 and 3600 seconds. This determines how often the Data Gateways screen in Control Room screen is refreshed.
6. Select the type of user that will be used by the Blue Prism database connection settings to connect to the Blue Prism database.
   - **Integrated security** – Windows authentication will be used to connect to the Blue Prism database. The account configured in the Data Gateways Settings tab on the Blue Prism server will be used.
   - **SQL authentication user** – SQL authentication will be used. Select the credential already added that contains the SQL credentials to be used. Only credentials that are configured as *Data Gateways Credentials* will be available for selection.
7. Update the default Port used for the Blue Prism database connection settings, if required. This is the port that Data Gateways will use to attempt to connect to the Blue Prism database. The default value is 1433.
8. Click **Apply** to save the settings.

Create a Data Gateways configuration

A Data Gateways configuration is a collection of outputs that define where data from session logs, published dashboards, and/or custom objects is sent. For each output, data can be sent to a file, HTTP endpoint, Splunk instance, or a database.

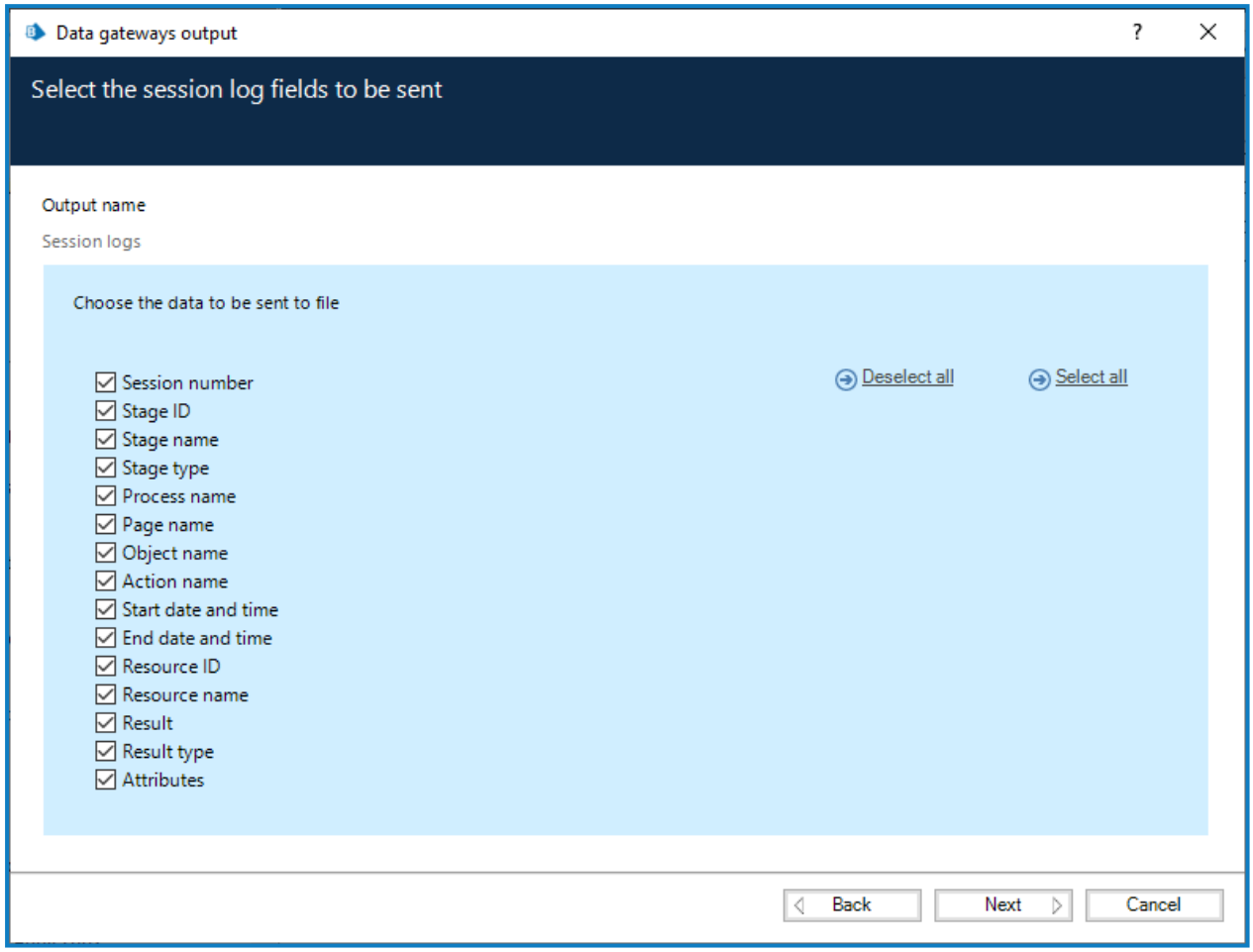
A Data Gateways configuration can consist of any number of individual outputs. The data from each of the outputs is added to a single configuration file.

1. Click the **System** tab and select **Data Gateways** > **Configuration**.
2. Click **Add new gateway output**. The Data gateway output wizard displays.



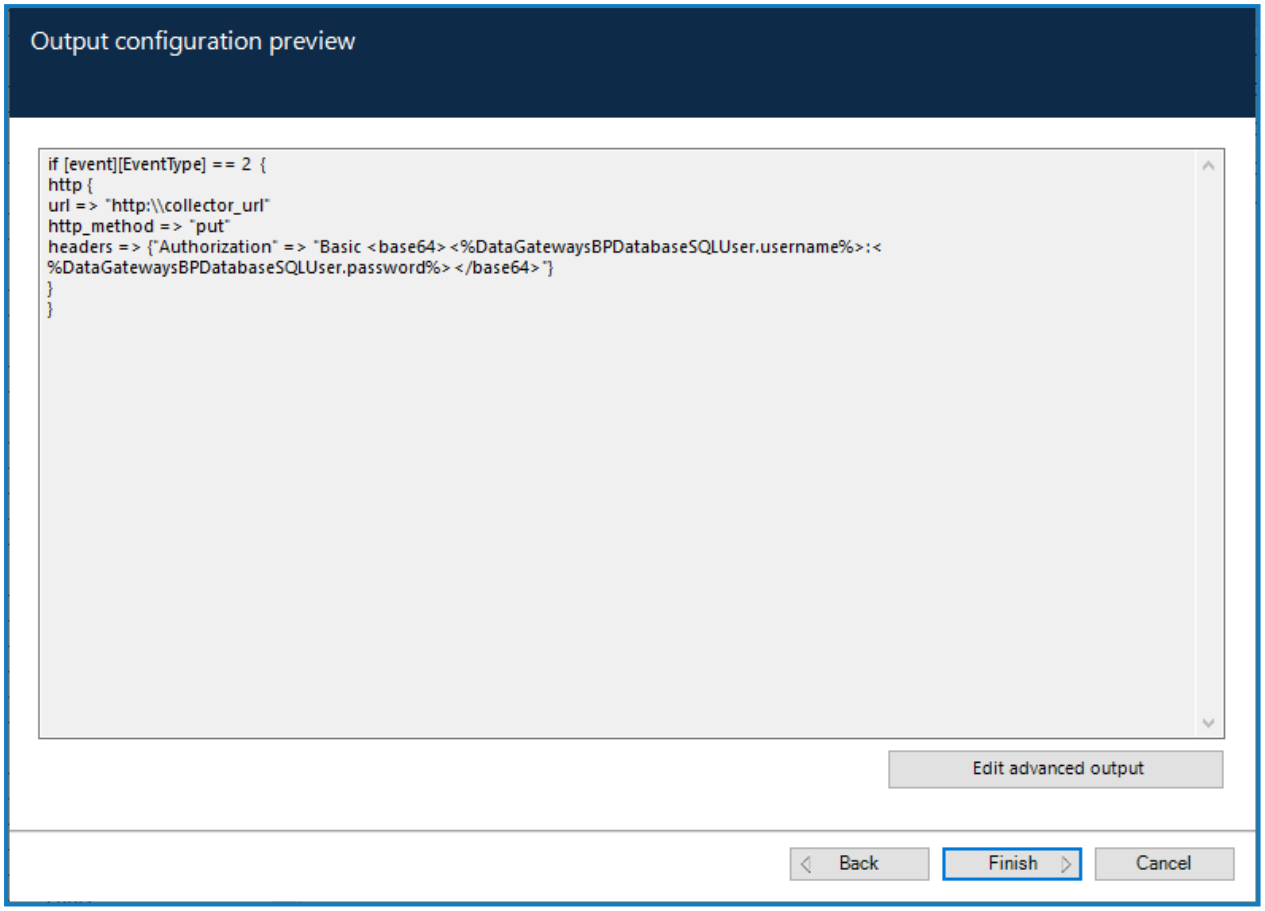
3. Enter a unique name for the output.
4. Select an output type, File, HTTP endpoint, Splunk, or Database and complete the fields for that output type.
5. Click **Next**.

6. Select the data to send to the configuration file for the specified output type:
   - **Session logs** – Data Gateways will process session log data for the specified output type. If you choose this option for any output type, select which session log data will be included in the output.



   - **Published dashboards** – Data Gateways will process data from the configured published dashboards for the specified output type.
   - **Work queue analysis snapshot data** – Data Gateways will process the work queue analysis data for the specified output type.
   - **Custom object data** – Data Gateways will process data from any Blue Prism action configured to use the Data Gateways internal business object.

7. Click **Next**. A preview of the output data displays.



The data in the produced output can be edited directly by clicking **Edit advanced output**. For more information about advanced configurations, see Advanced outputs.

8. Click **Finish** to save the output to the configuration.
9. A Data gateways message displays, prompting you to restart Data Gateways to apply any changes. Click **OK** to close the message. For more information on restarting the data gateway process see Start and stop the Data Gateways engine.

Configure Data Gateways for SQL authentication

Prerequisites

The following conditions must be met before configuring Data Gateways:

- The Blue Prism version must be version 6.5 Enterprise Edition or above, which must be configured to connect via at least one application server.
- Data Gateways requires a dedicated SQL server login with access to the Blue Prism database, created in SQL server. Whether using SQL or Windows authentication a user must exist with access to the Blue Prism database using the BPA_DataGateways role.
- The user account that runs the Blue Prism application server process must have read and write permissions to the file directory where the Logstash configuration file (bpconf.config) is stored. By default this is found in C:\Program Files\Blue Prism Limited\Blue Prism Automate.
- Data Gateways must be run as a single instance on an application server. Multiple instances cannot run on the same application server at the same time.

  If failover capacity is required in the case of the application server becoming unavailable, a secondary Data Gateways instance can be set up on a separate application server in the environment and manually started. After the new Data Gateways instance has started, the initial instance will disconnect and the most recently started instance will take over.

Authentication options for SQL login

You need to create a SQL login to access the Blue Prism Data Gateways database (this is described in the relevant authentication method below). You can then use one of the following authentication methods:

- Windows authentication (integrated security) using either existing or dedicated (recommended) credentials. If using existing Windows authentication credentials, the Data Gateways process will run in the same user context as the Blue Prism application server service. See Configure Data Gateways for Windows authentication for details if this is your chosen authentication method.
- SQL authentication, described in this section, using credentials configured via Blue Prism Data Gateways configuration. The service will run under the context of the Data Gateways service.

Install the Data Gateways engine components

Data Gateways requires a number of components to be installed and configured in the environment in which application servers are running the Data Gateways processes. The Data Gateways components are included in a separate installer, *Blue Prism Data Gateways Components*, available from the Blue Prism portal, see https://portal.blueprism.com/product/extras.

Create a Data Gateways login using a SQL authentication user

Data Gateways requires a SQL login, and a Windows authentication or SQL user to be created against the Blue Prism database. This section describes how to create a SQL login and a Windows authentication user.

Use Microsoft SQL Server Management Studio to create a SQL login using SQL server authentication. Alternatively, run the SQL script below.

Create SQL login and user

If you plan to use SQL authentication on your database, you can run the following SQL query against the server instance on which your Blue Prism databases are located. This query creates a login and user that allows Data Gateways to access the server.

> Only users with ALTER ANY LOGIN permission on the server, or membership to the securityadmin or sysadmin fixed server roles can create logins.

To create a SQL login and user, run the following command:

```
CREATE LOGIN [loginname] WITH PASSWORD = 'password';
CREATE USER [username] FOR LOGIN [loginname];
GO
sp_addrolemember 'BPA_DataGatewaysEngine', '[username]'
GO
```

Before running the query, enter the required Blue Prism database and replace *password* with a complex password for the Data Gateways login. For example:

```
CREATE LOGIN [Data Gateways SQL Login] WITH PASSWORD = '&CTh76£NkM';
```

Replace *username* with a user name of your choice and replace loginname with SQL login credentials name. For example:

```
CREATE USER [Data Gateways User] FOR LOGIN [Data Gateways SQL Login];
GO
sp_addrolemember 'BPA_DataGatewaysEngine', '[Data Gateways SQL User]'
GO
```

Enable the Data Gateways process

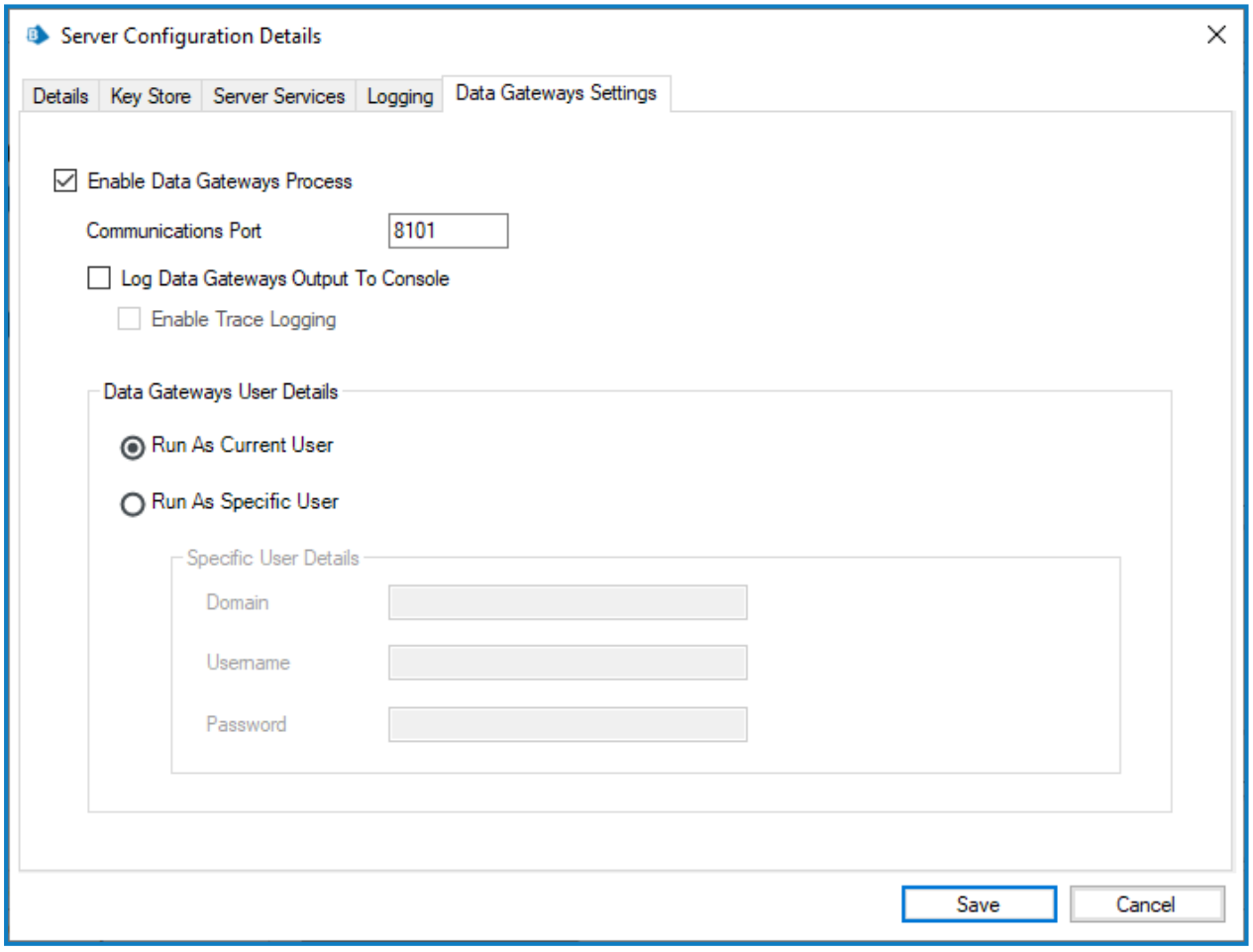
This section describes how to enable Data Gateways in BP Server and applies to SQL authentication and Windows authentication users.

The Data Gateways process must be enabled and the associated port defined on all application servers that are required to run Data Gateways. The Data Gateways process can only run on one instance on a single device. Where multiple application servers are configured on a single device, the Data Gateways process must be enabled on each one.

1. Open BPServer.exe and open the server configuration for the relevant environment.
2. Click the **Data Gateways Settings** tab.
3. Select **Enable Data Gateways Process** and enter the port that the application server will listen on for commands to start or stop the Data Gateways process.

   This port is also used by other Blue Prism application servers within the same environment to control the Data Gateways engine. This can be left as the default port (8101) unless there is another process on the application server that is already using that port. Ensure that security systems, such as firewalls, do not prevent other application servers from communicating with the Data Gateways engine via this port.

> The application server must be stopped before changing the port number.



4. If required, select **Log Data Gateways Output To Console** to send high-level Data Gateways messages to the console log in BPServer. This data can be used to help diagnose any potential issues with the Data Gateways integration. This option should not be left selected when not specifically required.
5. Select **Enable Trace Logging** to enable verbose Data Gateways logging, which can be used to further diagnose any issues between Blue Prism and Data Gateways.
6. In Data Gateways User Details, select one of the following options:
   - **Run as Current User** – The Data Gateways engine will operate in the same context as the Blue Prism Server service. Select this option if the Data Gateways engine will log into the Blue Prism database using a SQL authentication account. The credentials to use will be specified separately via the Data Gateways Configuration.
   - **Run as Specific User** – The Data Gateways engine will operate under the context of the Windows or Network account provided. Select this option if the Data Gateways engine will log into the Blue Prism database using a Windows Authentication account. It will access SQL using the specified credentials. If you select this option, additional configuration is required on the application server running the Data Gateways engine to configure a temporary folder that the Logstash components running under this context can access. See Additional configuration required to 'Run as Specific User for details.

It is not recommended to use *Run as Current User* to access the Blue Prism Database using Windows authentication. This configuration will result in the database being accessed using the account that is running the Blue Prism Server service which could result in the Data Gateways functionality having greater access to the database than it should have.

If using integrated security, we recommend using your own certificate to encrypt the Blue Prism server configuration files to make the configuration data more secure. This can be configured via *Manage Config Encryption Settings* in BPServer.exe.

9. Click **Save** to apply the settings.
10. Start the Blue Prism Server Service from the Windows Services menu.

Additional configuration required to 'Run as Specific User'

If you have configured Data Gateways to *Run as a Specific User* you will need to perform the following steps to configure a temporary folder that the Logstash components running under this context can access. Typically *Run as a Specific User* is selected when you require the Data Gateways engine to authenticate against the database using Windows authentication.

The following steps must be performed on the application server running the Data Gateways engine:

1. Create a new temporary directory on the C:\ drive, for example *dg-user-access*.
2. Assign the default *Authenticated Users* permissions to the directory.
3. Open the *C:\Logstash\logstash\config\jvm.options* file in a text editor.

4. Add the temporary directory's path to the `-Djava.io.tmpdir=` line. Uncomment the line if it is commented out (remove the # from the beginning of the line).

```
## basic

# set the I/O temp directory
-Djava.io.tmpdir=C:/dg-user-access

# set to headless, just in case
-Djava.awt.headless=true
```

5. Save the changes.
6. Restart the Data Gateways service or the application server to apply the updates.

Create SQL Data Gateways credential in Blue Prism

Data Gateways credentials are a type of credential record that can be used when configuring Data Gateways. They can be used to store the credentials needed to access databases or HTTP end points. In addition to this, a Data Gateways credential must be created to store the SQL server user account that provides the Data Gateways engine with access to the Blue Prism database, if using SQL (versus Windows) authentication.

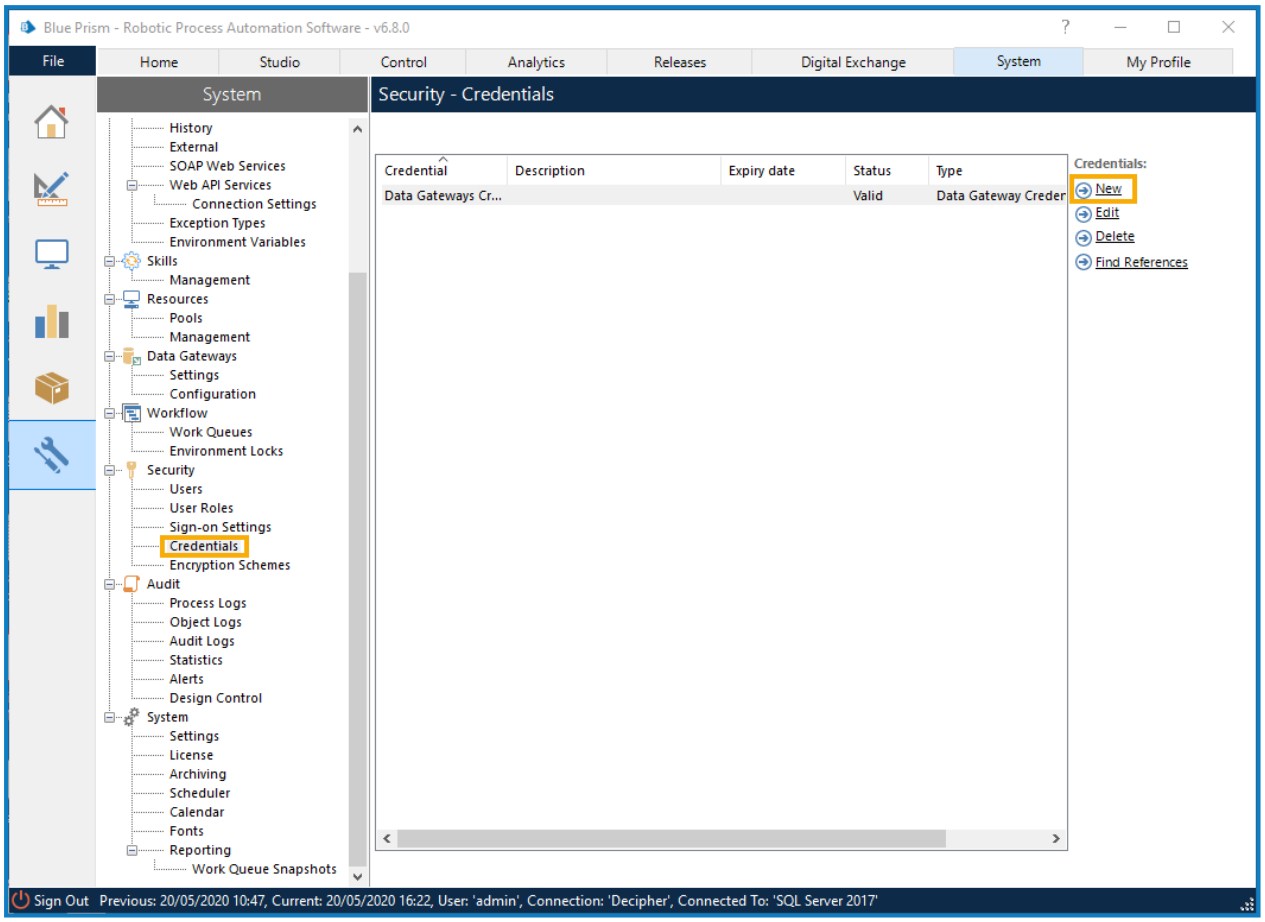
Credentials that use the Data Gateways credential type are only accessible by the application server when generating Data Gateway configurations – they cannot be used in standard Blue Prism processes or other Blue Prism elements such as web API definitions.

Add a credential for the Blue Prism database SQL server user

Create a credential to store the SQL Server user account required for secure access to the Blue Prism database.

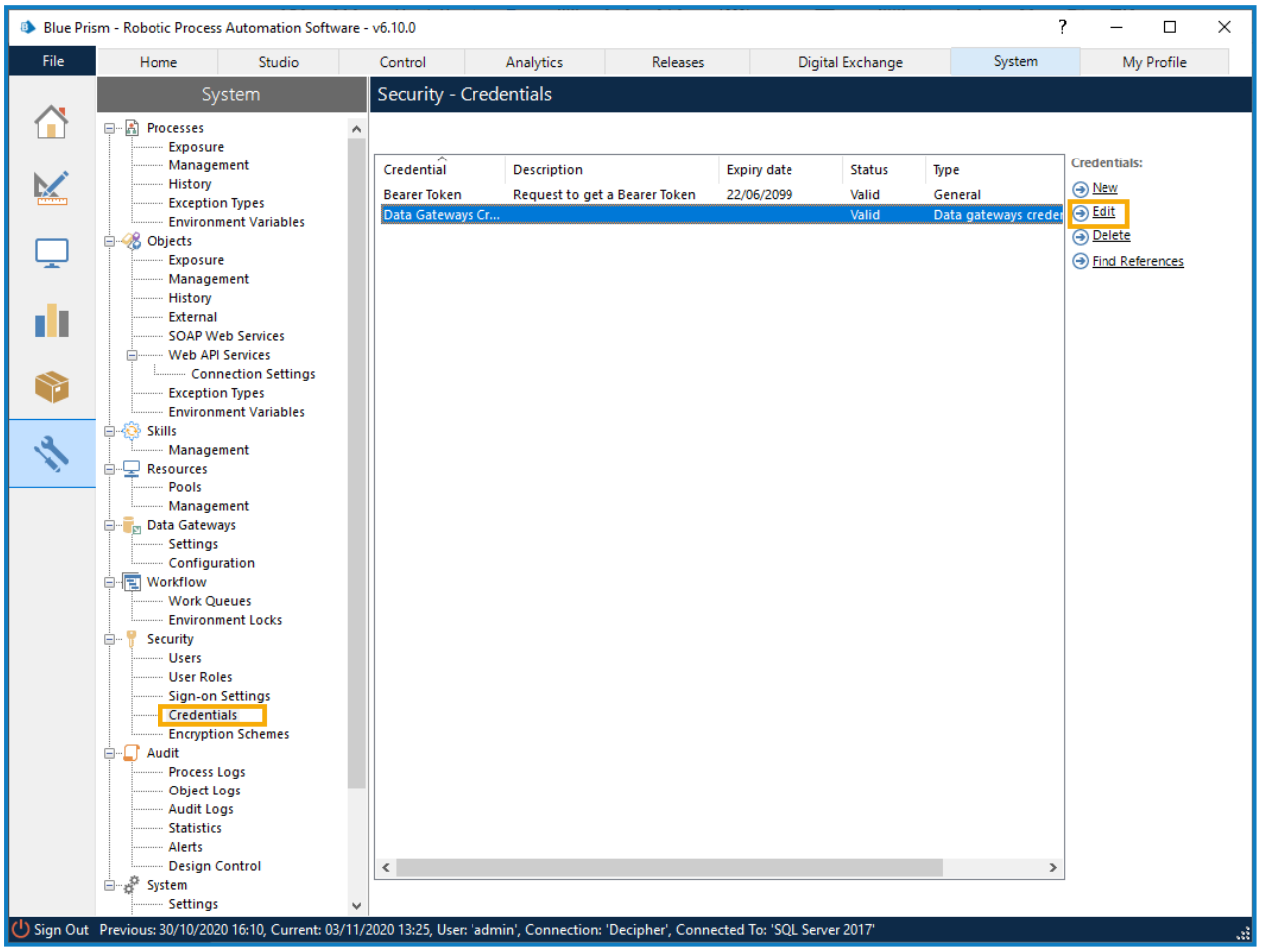
> The credential name, username, and password must exactly match those already created for the SQL Server login.

1. Select the **System** tab.
2. Select **Security** > **Credentials**.



3. Click **New**. The Credential Details window displays.

4. Configure the following credential:
   - **Name** – Enter the SQL server login name that you have created.
   - **Type** – Select *Data Gateway Credential*.
   - **Username** – Enter the user name that you have created for the SQL server login.
   - **Password** – The password that is specified for the SQL server user.



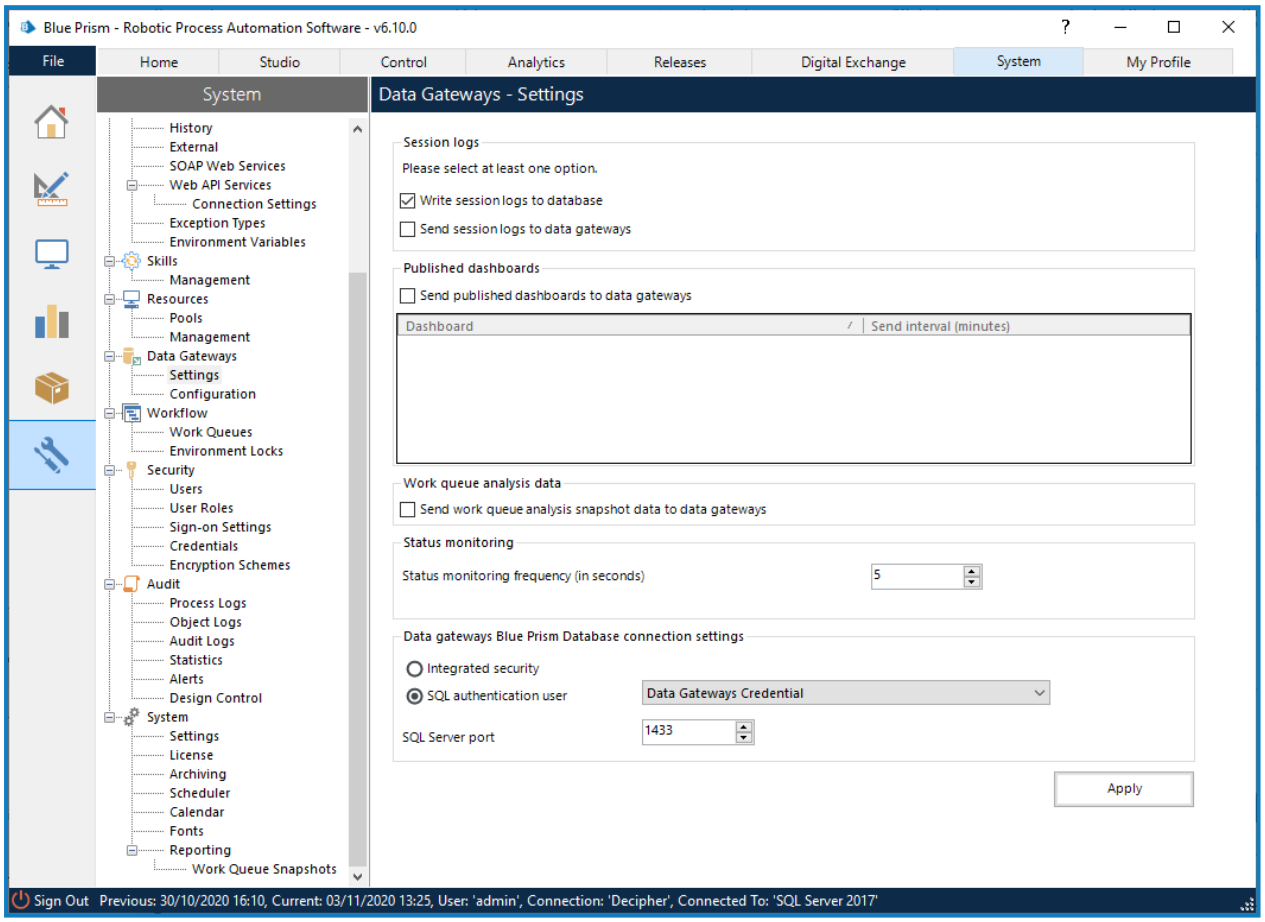
5. Click **OK** to save the credential.

If you attempt to delete the Data Gateways credential that is currently in use a warning message displays and you are prompted to confirm the deletion.

Configure Data Gateways settings

Configure the Data Gateways engine by determining what data will be processed and where relevant, the frequency at which data is sent.

1. From the **System** tab, select **Data Gateways** > **Settings**.



2. Select the required options to determine where session log data will be stored - at least one option must be selected:
   - **Write session logs to database** – Session logs will be sent to the session log table in the Blue Prism database. If not selected, the functionality to view new session logs in the Blue Prism client will be unavailable. This is the default setting and should only be changed if you no longer want new session log records to be accessible via the Blue Prism user interface.
   - **Send session logs to data gateways** – Session logs will be sent to a temporary storage table on the Blue Prism database where they are accessed by the Data Gateways engine for use in the configured outputs. The logs will be deleted from the temporary table after they have been processed by Data Gateways.

     If the specified HTTP, Splunk, or Database Data Gateway endpoint is not reachable when the data is being processed, the data is stored temporarily until the endpoint becomes available. See Data Storage when endpoint unavailable for details.

3. Select **Send published dashboards to data gateways** to send data from published dashboards to a database table in the Data Gateways system and set the frequency that data is sent for each dashboard. For more information about Blue Prism dashboards, see Dashboards.

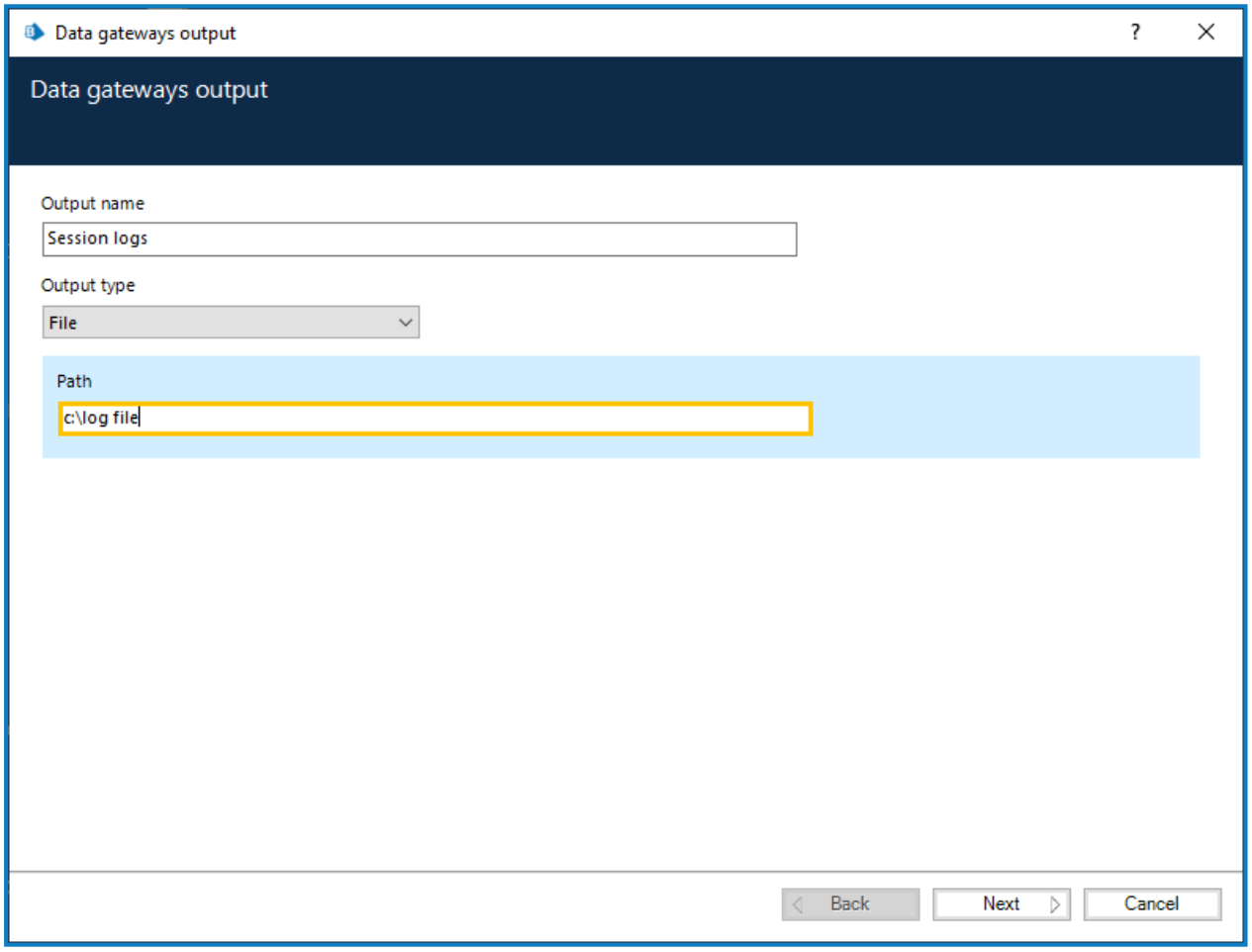
4. Select **Send work queue analysis snapshot data to data gateways** to send work queue analysis data to the database. For more information about Blue Prism work queue analysis snapshots, see the Work queue snapshots topic in the in-product help.
5. Set the **Status monitoring frequency** to a value between 5 and 3600 seconds. This determines how often the Data Gateways screen in Control Room screen is refreshed.
6. Select the type of user that will be used by the Blue Prism database connection settings to connect to the Blue Prism database.
   - **Integrated security** – Windows authentication will be used to connect to the Blue Prism database. The account configured in the Data Gateways Settings tab on the Blue Prism server will be used.
   - **SQL authentication user** – SQL authentication will be used. Select the credential already added that contains the SQL credentials to be used. Only credentials that are configured as *Data Gateways Credentials* will be available for selection.
7. Update the default Port used for the Blue Prism database connection settings, if required. This is the port that Data Gateways will use to attempt to connect to the Blue Prism database. The default value is 1433.
8. Click **Apply** to save the settings.

Create a Data Gateways configuration

A Data Gateways configuration is a collection of outputs that define where data from session logs, published dashboards, and/or custom objects is sent. For each output, data can be sent to a file, HTTP endpoint, Splunk instance, or a database.

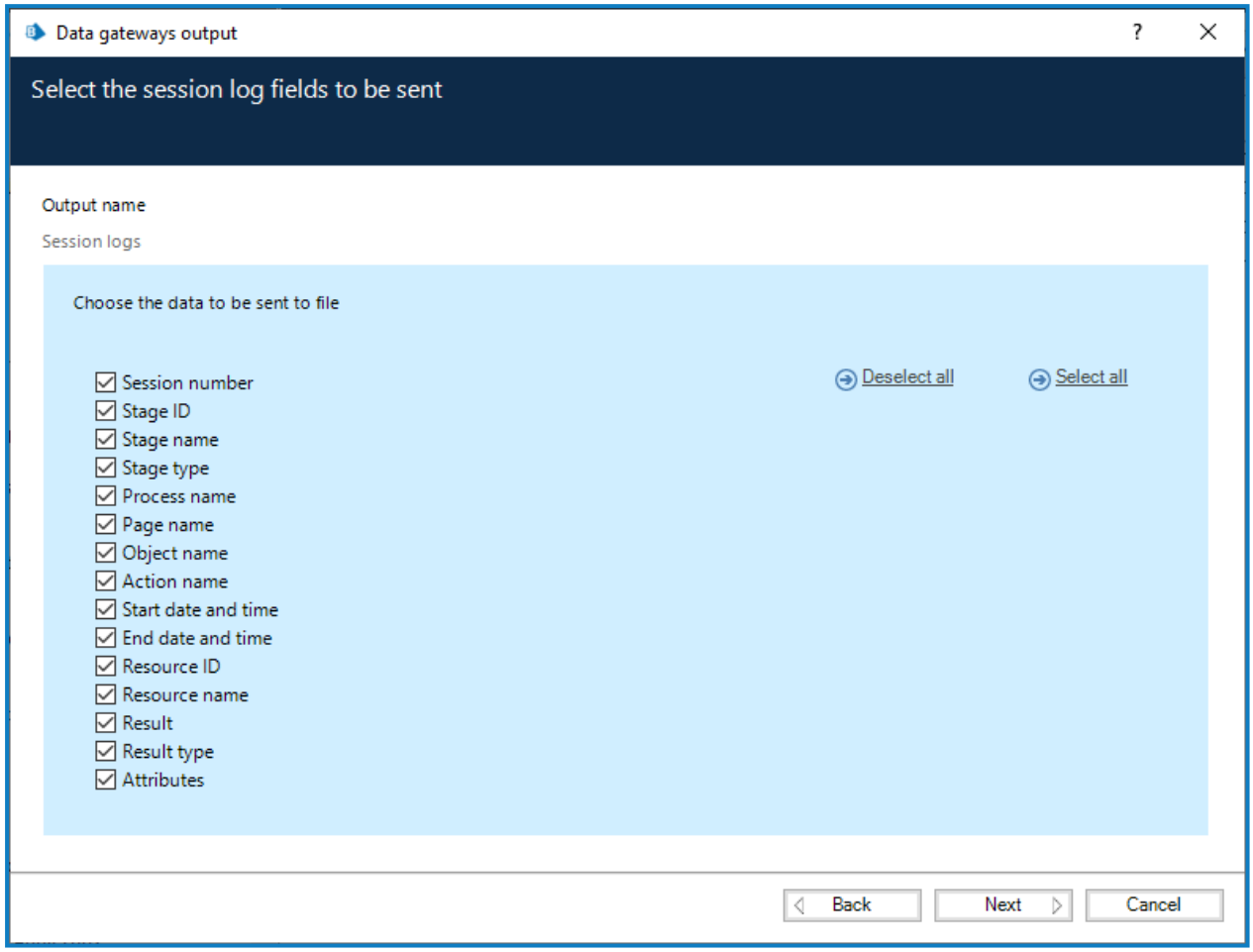
A Data Gateways configuration can consist of any number of individual outputs. The data from each of the outputs is added to a single configuration file.

1. Click the **System** tab and select **Data Gateways** > **Configuration**.
2. Click **Add new gateway output**. The Data gateway output wizard displays.



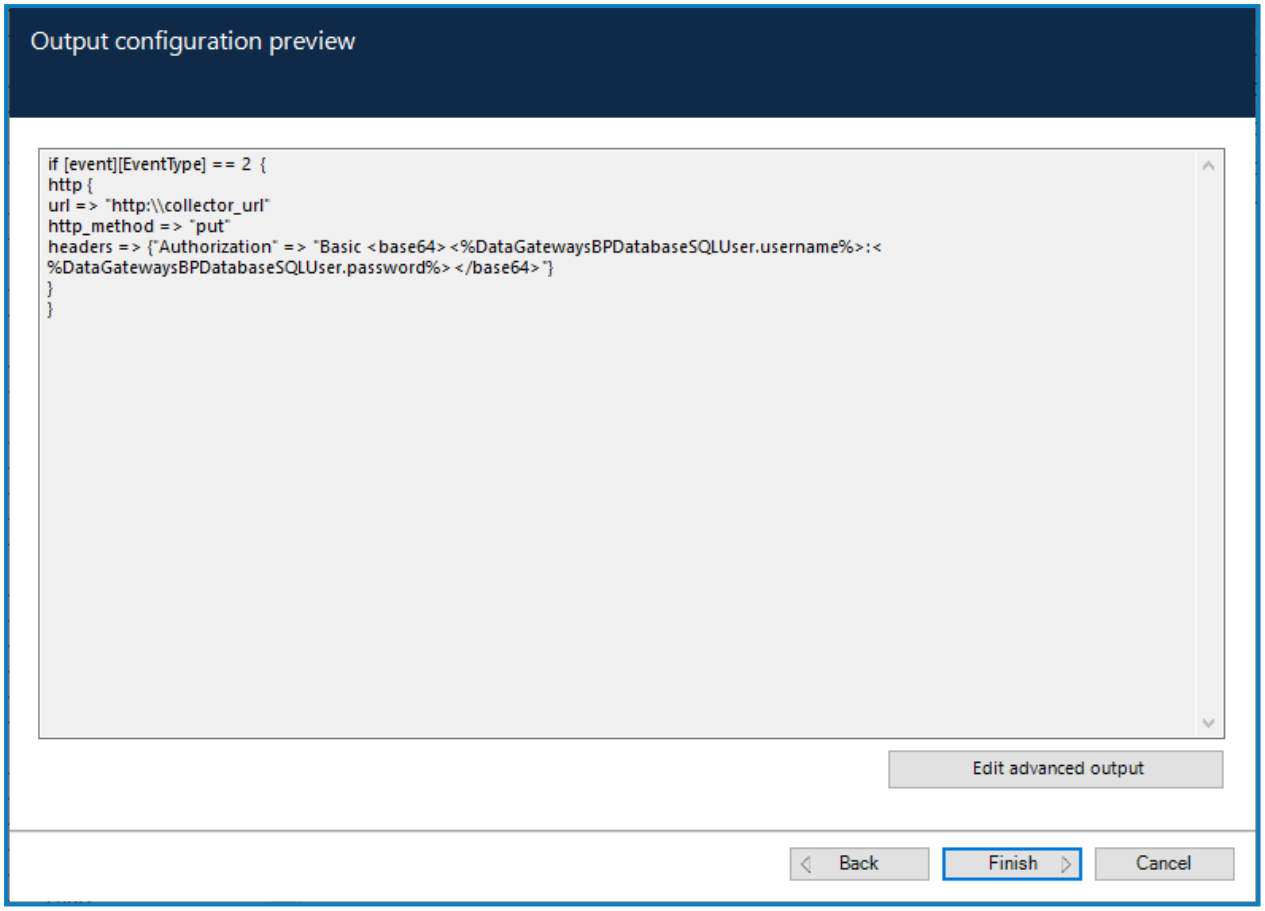
3. Enter a unique name for the output.
4. Select an output type, File, HTTP endpoint, Splunk, or Database and complete the fields for that output type.
5. Click **Next**.

6. Select the data to send to the configuration file for the specified output type:
   - **Session logs** – Data Gateways will process session log data for the specified output type. If you choose this option for any output type, select which session log data will be included in the output.



   - **Published dashboards** – Data Gateways will process data from the configured published dashboards for the specified output type.
   - **Work queue analysis snapshot data** – Data Gateways will process the work queue analysis data for the specified output type.
   - **Custom object data** – Data Gateways will process data from any Blue Prism action configured to use the Data Gateways internal business object.

7. Click **Next**. A preview of the output data displays.



The data in the produced output can be edited directly by clicking **Edit advanced output**. For more information about advanced configurations, see Advanced outputs.

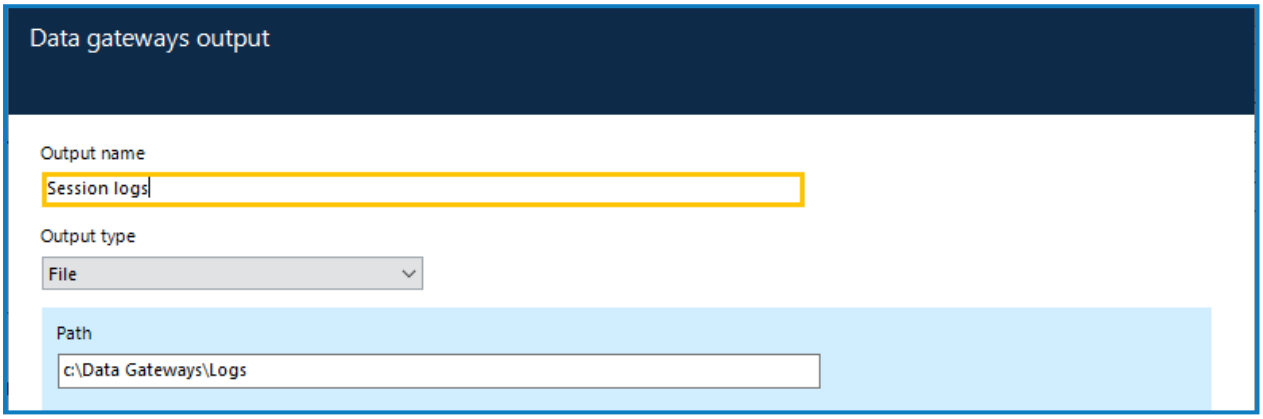
8. Click **Finish** to save the output to the configuration.
9. A Data gateways message displays, prompting you to restart Data Gateways to apply any changes. Click **OK** to close the message. For more information on restarting the data gateway process see Start and stop the Data Gateways engine.

Data Gateways output types

File

Export the selected data to the specified file. Although creating outputs in the Data Gateways Configuration wizard only supports sending data to .txt files, advanced outputs and custom configurations can be edited to support other file formats, such as .csv.

For details about how to achieve this, see Data Gateways configuration files.



Dynamic file paths can be used to include the creation date in filenames using variables for year, month, day, and time.

For example: C:\Session Logs\session_logs_%{+YYYY-MM-dd}.txt

This will create a new text file each day in the specified location, that includes the current date in the filename: session_logs_2019-04-30.txt

The variables can also be used in file paths to create date-based folder names. For example:

C:\Session Logs\%{+YYYY}\%{+MM}\%{+dd}\session_logs_%{+YYYY-MM-dd-hh}.txt

This will create folders based on the current date and append the filename. For example:

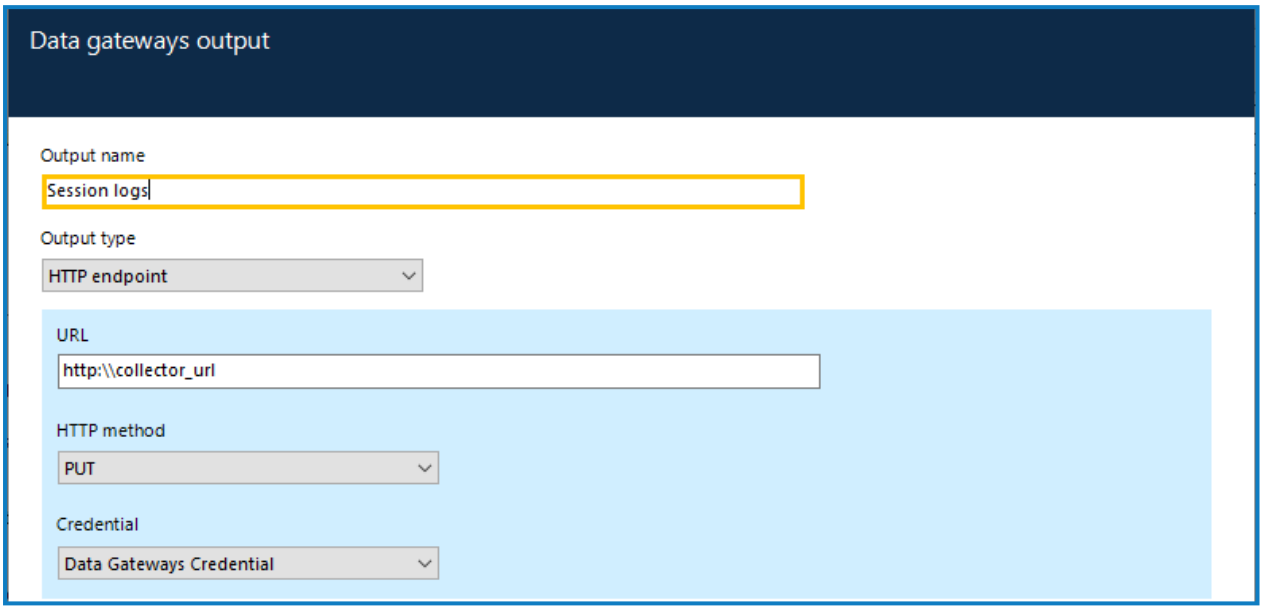
C:\Session Logs\2019\04\30\session_logs_2019-04-30-11.txt

The file/folder timestamp generated when using dynamic file paths is in UTC format.

Dynamic file paths can be used in configurations created using the data gateways wizard and advanced configuration.

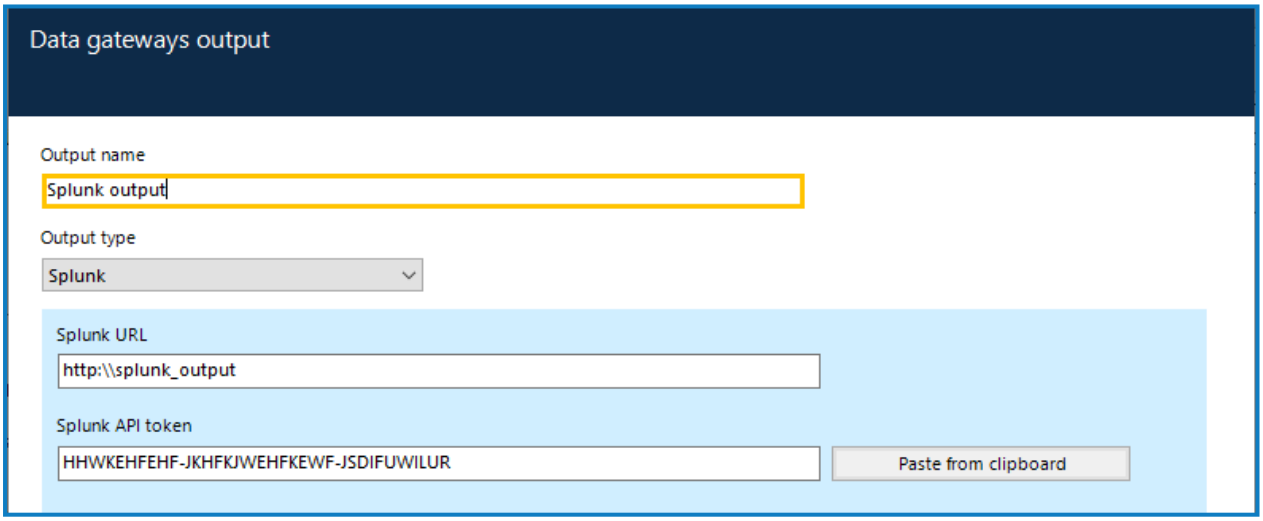
HTTP endpoint

Route data through the specified HTTP endpoint using the selected method. Select a Data Gateways credential if authentication is required to access to the endpoint.



Splunk

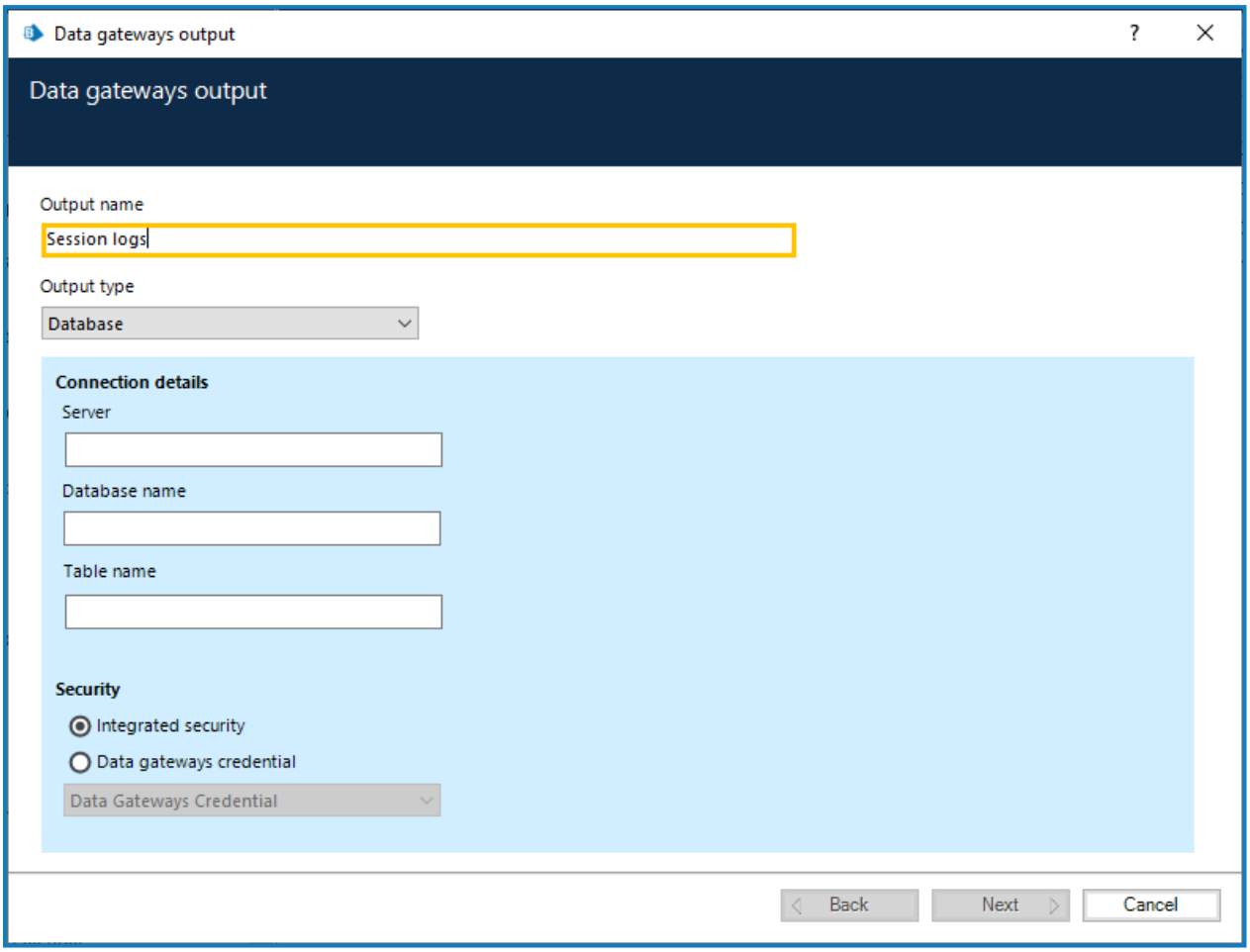
Send the selected data to the specified Splunk URL using the associated API token.



To successfully configure a Splunk output, the HTTP event collector must be enabled in your Splunk configuration. For further details, see https://docs.splunk.com/Documentation/Splunk/7.3.1/Data/UsetheHTTPEventCollector.

Database

Send the selected data to the specified SQL server database table. Select a Data Gateways credential if authentication is required to access to the database.



The security options relate to the target output database and not the Blue Prism Database.

Integrated Security – Data Gateways will connect to the target database using Windows authentication. It will use the account that the Data Gateways engine runs under – this is configured using the following options in the BP Server configuration:

- If *Run as Current User* is selected, it will connect to the target database using the same context that the Blue Prism Server runs as, using Windows Authentication. This means that the user that the Blue Prism Server runs as, will need to be granted permissions on the target system.
- If *Run as Specific User* is selected, it will connect to the target database using the credentials specified on the Blue Prism server, using Windows authentication. This means that the user configured as the Data Gateways user on the Blue Prism Server configuration will need to be granted permissions on the target system.

Data gateway credential – Data Gateways will connect to the target database using the credentials contained within the specified Credential record. It will authenticate against the database using SQL authentication.

Target output database

The target output database:

- Should be a new, separate database or database server.
- Cannot be an existing Blue Prism database.
- Should contain a single table, with a recommended name of `DGOutput`.

> The table name must not be named `BPADataPipelineInput`.

The specified database table must contain the following columns:

- An eventType column of type *integer* to store the event type.
- An eventData column of type *nvarchar(max)* to store the events serialized to a JSON string.

Optionally, the database table can also include the following column:

- An insertedDate column of type *datetime* with a default value of GETUTCDATE() to store the date when the record was added to the table. You can use this column to clean up data in the table.

For example, to create the table you can use the following table definition:

```
CREATE TABLE [dbo].[DGOutput](
	[eventType] [int] NULL,
	[eventData] [nvarchar](MAX) NULL,
	[insertedDate] [datetime] NOT NULL CONSTRAINT [DF_DGOutput:insertedDate]  DEFAULT
(GETUTCDATE())
) ON [PRIMARY]
GO
```

Example queries

You can query the data in the target output database. For example:

```
SELECT [eventType],[eventData]
FROM DGRepository.dbo.DGOutput;
```

eventType can be:

- 1 – Session log
- 2 – Published dashboard
- 3 – Custom
- 4 – Work queue analysis

eventData is stored as a JSON string of key value pairs.

For example, to return all the session log records, you could use the following query:

```
SELECT [eventType],[eventData]
FROM DGRepository.dbo.DGOutput
WHERE eventType = 1;
```

If the table resides on a SQL Server instance running either SQL Server 2016 or later, Azure SQL Database, or Azure SQL Managed Instance, you can filter the eventData column using the JSON_VALUE () function. The following query returns all session log records with a StartDate greater than 01/01/2020:

```
SELECT [eventType],[eventData]
FROM DGRepository.dbo.DGOutput
WHERE eventType = 1
AND JSON_VALUE([eventData], '$.StartDate') > '2020-01-01T00:00:00+00:00';
```

You can also use date manipulation in your queries. The following query converts the StartDate to a Date data type, and return all records older than 183 days:

```
SELECT [eventType],[eventData]
FROM DGRepository.dbo.DGOutput
WHERE eventType = 1
AND CONVERT(DATE,JSON_VALUE(eventData, '$.StartDate')) < GETDATE()-183;
```

You can also include wildcard searches. The following query returns records where the value of ResourceName is like BPEU5:

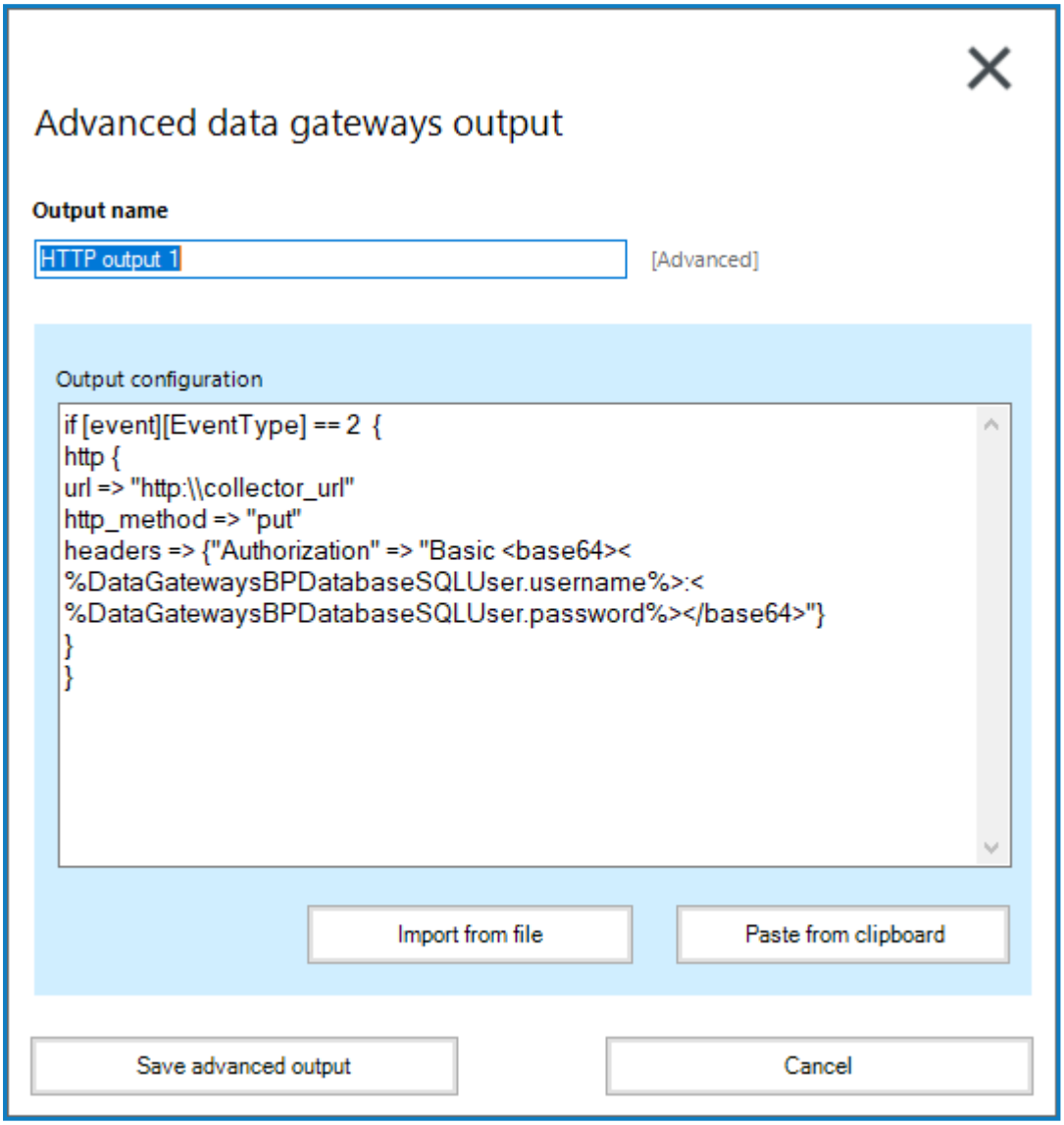
```
SELECT [eventType],[eventData]
FROM DGRepository.dbo.DGOutput
WHERE eventType = 1
AND JSON_VALUE([eventData], '$.ResourceName') LIKE '%BPEU5%';
```

To use a port other than the default port, the server name can be appended with an alternative port number, separated by a colon. For example, *bp-server-1:8001*.

Data Gateways advanced outputs

When creating or editing an output, users with the *Data Gateways – Advanced Configuration* permission can edit the code in the output directly to create an advanced configuration.

From the output configuration preview screen of the output wizard, click **Edit advanced output** to open the advanced editor.

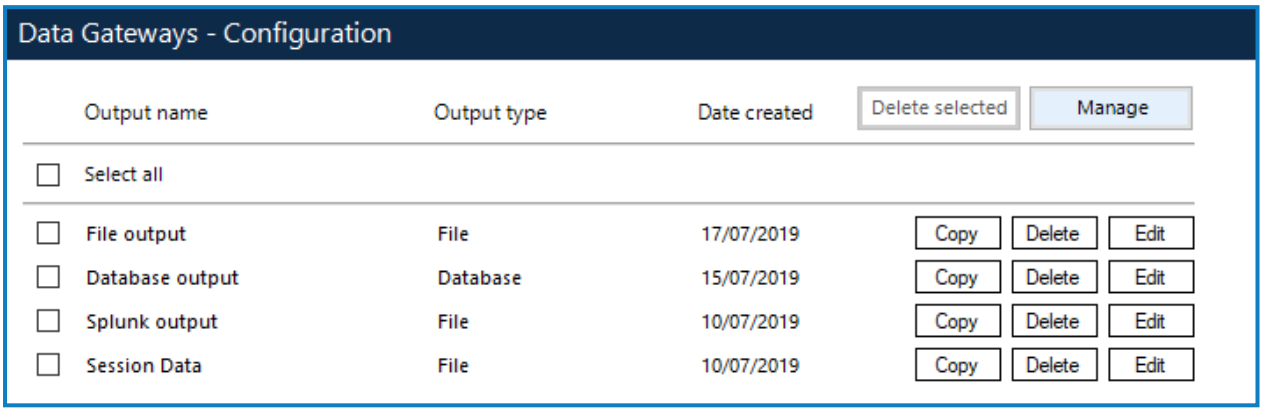


The output can be edited by entering data or pasting it from the clipboard. Files up to 10k in size can also be imported into an output, replacing all current data. When the required changes have been made, click **Save advanced output**.

Once an output has been saved as an advanced output, even if no changes have been made, it can no longer be configured in the Data Gateway output wizard – only in the advanced editor.

Manage a Data Gateways configuration

The outputs for a Data Gateways configuration are listed in the **Data Gateways** > **Configuration** screen. Click **Manage** to display the options to copy, edit, or delete one or more configurations.



Every time a Data Gateways configuration is changed, whether that is new, updated, or deleted output, the Data Gateways engine must be restarted. For more information, see Start and stop the Data Gateways engine.

Copy an output

Click **Copy** for the output you want to duplicate. Depending on the output type, the Data Gateways wizard or advanced editor will open enabling the duplicated output to be updated as required. When the output has been saved, restart the Data Gateways process.

Delete an output

Click **Delete** for the required output to remove it from the configuration and restart the Data Gateways process.

Edit an output

Click **Edit** for the required output. Depending on the output the output wizard or advanced editor displays allowing the output to be updated. When the output has been saved, restart the Data Gateways process.

Delete selected output

Click the **Select all** check box in the Output name column to select all of the outputs for group deletion. Select/deselect the check box for a specific output, as required. Click **Delete selected** to delete all selected output(s).

Data Gateways custom configurations

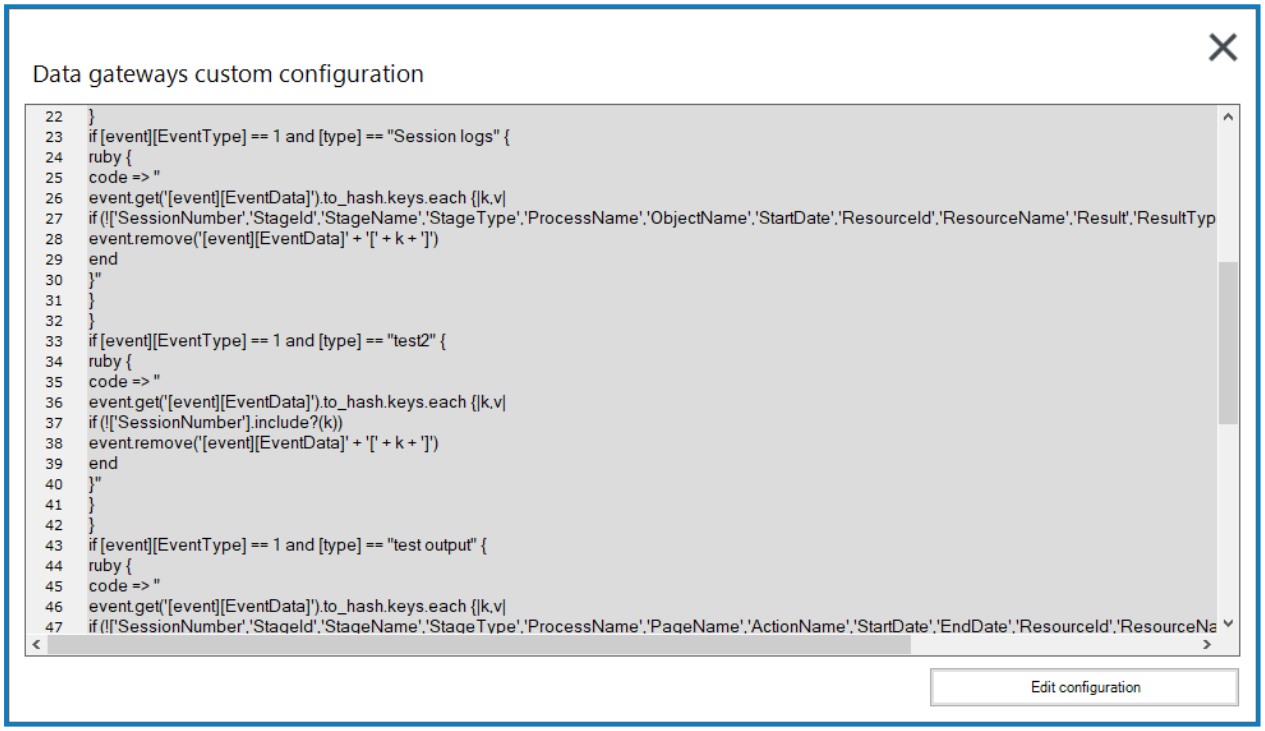
Users with the *Data Gateways – Advanced Configuration* permission can create a custom configuration by directly editing the underlying code. This allows users to create a configuration that has functionality beyond what is available through the wizard such as the ability to only send specific event fields to an output or customizing the format of the data that is sent to an output.

For further information about the composition of configuration files, see Configuration file structure.

Create a custom configuration

1. From the *Data Gateways – Advanced Configuration* click **View configuration**

   The configuration code displays.

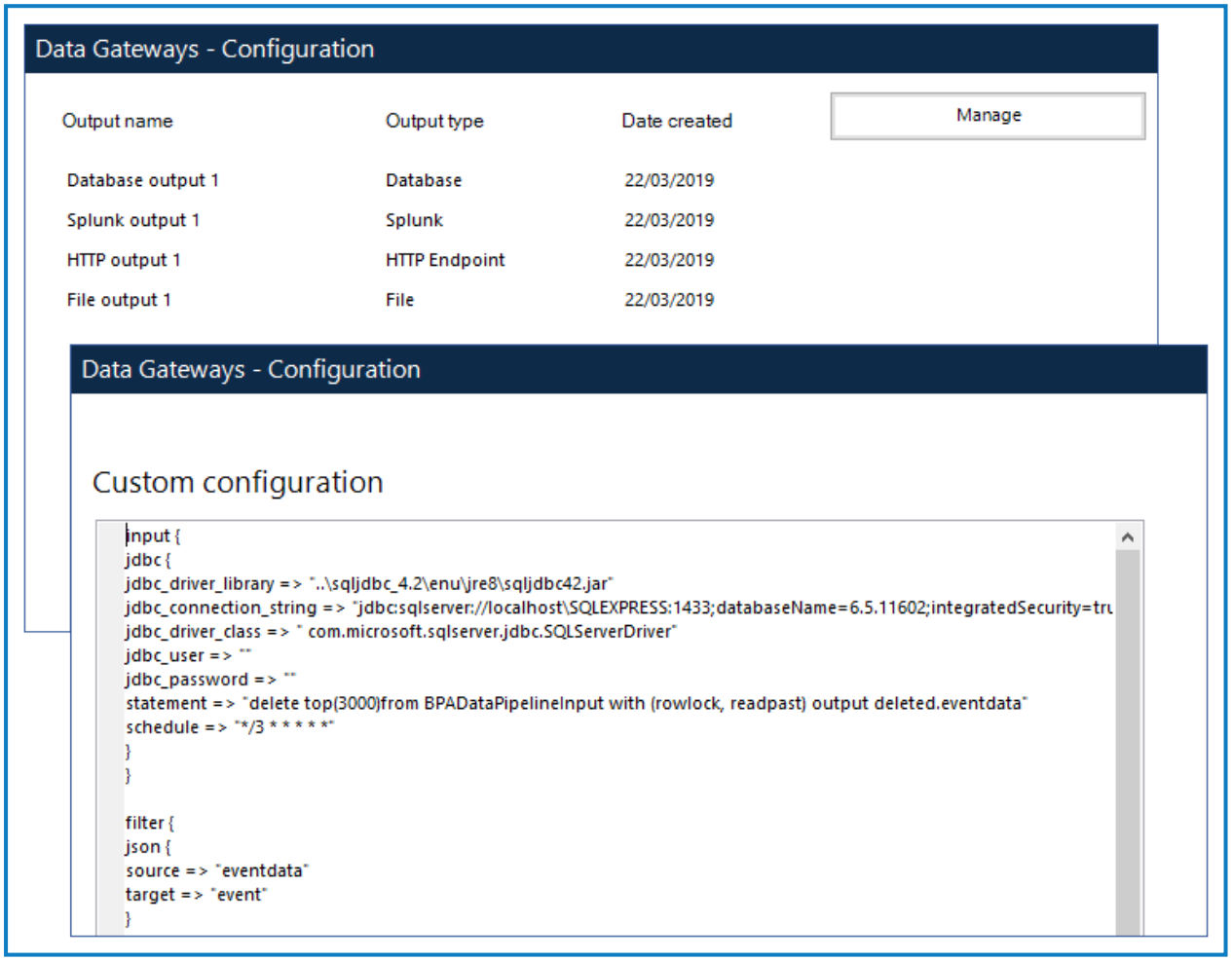


2. Click **Edit configuration** to make the configuration code editable.

3. When the required updates have been made, click **Save** and confirm.

> The Data Gateways engine must be restarted for the changes to take effect. For more information, see Start and stop the Data Gateways engine.

The custom configuration overwrites the outputs in the existing one. This is reflected in the Data Gateways screen, where the advanced configuration code replaces the list of outputs.



Click **Delete** in a custom configuration at any time to revert the configuration to its original composition and display the original outputs.

Manage Data Gateways processes in Control Room

Data Gateways processes are listed in Control Room from where the status can be viewed and the process restarted.

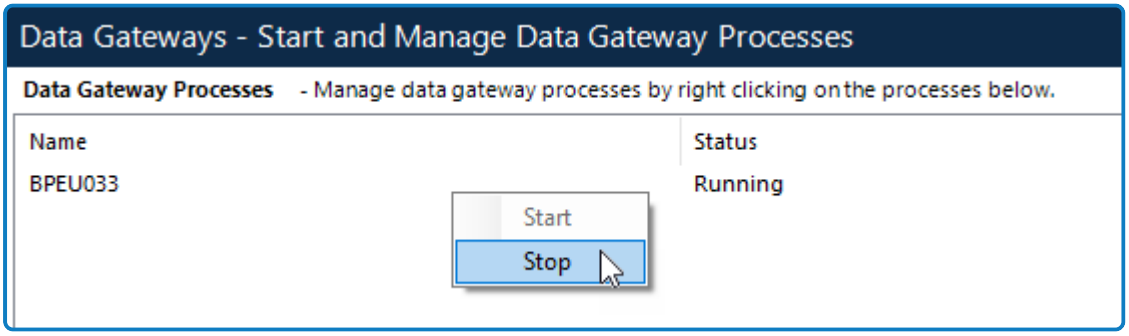
Select the **Control** tab and click **Data Gateways** to display all the Data Gateways processes in the environment.

Processes can be in the following states:

- **Online** - The application server hosting the Data Gateways process is online.
- **Offline** - The application server hosting the Data Gateways process is offline.
- **Starting** - The Data Gateways process is starting up.
- **Running** - The Data Gateways process is processing events.
- **Error** - The Data Gateways process has encountered an error. It will attempt to restart periodically to rectify the error.
- **Unrecoverable Error** - The Data Gateways process has encountered an error that restarting will not fix. It will not attempt to restart.

Start and stop the Data Gateways engine

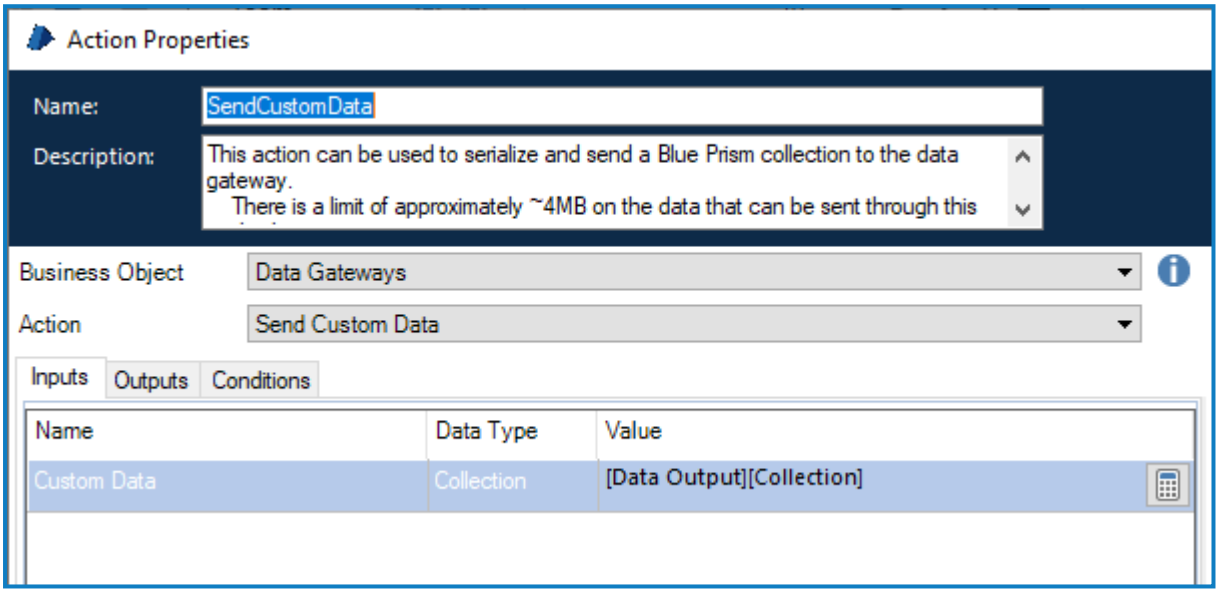
Right-click on a process to open the context menu. Depending on the current state of the process, select **Start** or **Stop** as required.



Data Gateways VBO

The Data Gateways internal business object can be used in action stages in any process and configured to send data from collections to the Data Gateways engine as the process runs. Any data that can be put into a collection can be sent to the Data Gateways engine using the VBO.

The VBO has one action, *Send Custom Data* that accepts collections as inputs and sends the data from those collections to the Data Gateways engine. The data from all actions that use the VBO will be used by any output that is configured to send custom object data.



Data Gateways user role permissions

There are three Data Gateways user permissions that can be granted to the appropriate administrator user roles:

- **Data Gateways – Configuration** – The user can configure Data Gateways settings and add and manage gateway outputs. They cannot create or edit advanced outputs or custom configurations.
- **Data Gateways – Advanced Configuration** – The user can configure Data Gateways settings and add and manage gateway outputs. They can create and edit advanced outputs and custom configurations. This permission should only be granted to expert users who have the knowledge to edit output and configuration code.
- **Data Gateways – Control Room** – The user has access to the Data Gateways node in the Control Room to check the status of the Data Gateways engine and start and stop the process as required.

These permissions are granted to the System Administrator for upgrades and new installations - they are not enabled for any other built-in user role.

Data Gateways configuration files

Configuration files are created from the configured Data Gateways settings and outputs. They are composed of three main sections, Input, Output, and Filter, as outlined below.

Input

This section is automatically generated by Blue Prism based on the Blue Prism database settings. It determines how events are pulled into the Data Gateways engine for processing. In the example below, they are retrieved from the BPADataPipelineInput table in the Blue Prism database.

```
input {
jdbc {
jdbc_driver_library => "..\sqljdbc_4.2\enu\jre8\sqljdbc42.jar"
jdbc_connection_string => "jdbc:sqlserver://SQL_SERVER_INSTANCE:1433;databaseName=BP_
DATABASE;"
jdbc_driver_class => "com.microsoft.sqlserver.jdbc.SQLServerDriver"
jdbc_user => "<%Data Gateways BP Database SQL User.username%>"
jdbc_password => "<%Data Gateways BP Database SQL User.password%>"
statement => "delete top(3000)from BPADataPipelineInput with (rowlock, readpast)
output deleted.eventdata"
schedule => "*/3* * * * *"
}
}
```

If required, the following areas of the input can be edited to suit the required preferences.

`schedule => "*/3* * * * *"`

This determines how often the SQL query to request data from the BPADataPipelineInput table is executed. The default value of every three seconds can be updated by replacing `3` with the required value.

`statement => "delete top(3000)from BPADataPipelineInput with (rowlock, readpast) output deleted.eventdata"`

This is the SQL statement that is executed against the Blue Prism database to pull events out of the BPADataPipelineInput table. The value controls the maximum number of rows pulled from the BPADatapipelineInput table every interval. The default value of 3000 can be changed if required.

Filter

Filters can be used to perform intermediary processing on an event. These could be actions such as adding, removing, or modifying certain fields of an event before they are sent to the outputs, for example, removing the AttributeXML field of a session log.

A list of all filter plug-ins available are listed here: https://www.elastic.co/guide/en/logstash/current/filter-plugins.html

By default, the configuration generated by Blue Prism will contains a single filter:

```
filter{
json {
source => "eventdata"
target => "event"
}
}
```

By default, configurations contain a single filter for JSON that is used to parse and expand the JSON string that contains the configured data types (session logs, dashboards etc.) so the contents are accessible in the configuration file.

This section can be edited to add and remove filters but the default JSON filter should not be removed or amended.

Output

Outputs determine where events are sent. If outputs have been configured using the Data Gateways wizard, they will be included here. Every event processed will be sent to every output listed in the configuration.

```
output {
file {
path => "C:\data.txt"
}

csv {
path => "C:\data.csv"
}
}
```

In the example above, a .txt file and a .csv output are specified. Every event sent to the Data Gateway system will be written into a text file at C:\data.txt and also a csv file at C:\data.csv

For a list of outputs available see here: https://www.elastic.co/guide/en/logstash/current/output-plugins.html

Event structure

This section details the structure of events in Logstash after they are received from Blue Prism. This information can be used to construct conditional statements in the Logstash configuration to divert events to outputs based on their content, or for creating custom message formats for your outputs.

The event (either session log or published dashboard) is stored in the Blue Prism database as a JSON string. In order to turn this JSON string into a set of fields which can be used in Logstash the JSON filter is added to the configuration:

```
filter{
json{source => "eventdata"
```

```
target => "event"}
}
```

This adds the session log / published dashboard as fields nested under the “event” field.

For example:

`[event][eventType]` contains the type of event (session log, published dashboard or custom object data).

`[event][EventData]` contains the data for the event as nested fields.

`[event][EventData][SessionNumber]` contains the Session number if this is a session log event.

To send only session logs from a process named “ProcessA” to a text file you a conditional statement around your output can be used:

```
output{
If [event][eventType] == 1 and [event][EventData][ProcessName] == “ProcessA” {
file {
path => “C:\log.txt”
}
}
}
```

For a full listing of the available fields see the following tables.

General

Event	Description
`[event][eventType]`	The number that represents the event type: 1 = Session Log 2 = Published Dashboard 3 = Custom 4 = Work Queue Analysis
`[event][EventData]`	The data for the event. The structure of this data will differ depending on the event type.

Event type – session logs

Event	Description
`[event][EventData][StartDate]`	The start date of the process stage formatted in ISO 8601 notation. For example: "2019-02-11T07:59:54.829674+00:00"
`[event][EventData][SessionNumber]`	The session number for the session this session log belongs to.
`[event][EventData][ResultType]`	The result type of the process stage.
`[event][EventData][Result]`	The result of the process stage.
`[event][EventData][AttributeXML]`	The input and output parameters of the stage serialized to XML.

Event	Description
[event][EventData][ProcessName]	The name of the process this stage belongs to. This will be empty if the session log is logged from a business object.
[event][EventData][ObjectName]	The name of the business object this stage belongs to. This will be empty if the session log is logged from a process.
[event][EventData][ActionName]	If this log is from an Action stage, this is the name of the Action. Otherwise it will be empty.
[event][EventData][PageName]	The name of the page this stage which created this session log belongs to.
[event][EventData][StageType]	The type of stage which created this session log.
[event][EventData][StageId]	The ID of the stage which created this session log.

Event type – published dashboards

Event	Description
[event][EventData][Source]	The name of the published dashboard.
[event][EventData][Subject]	The name of the dashboard tile which generated the data.
[event][EventData][Values]	The data from the dashboard tile.

Event type – custom object data

Event	Description
[event][EventData][CustomDataCollection]	The custom data from the process that will be sent.
[event][EventData][SessionNumber]	The session number of the process the data is coming from.
[event][EventData][StageID]	The ID of the stage that this action is called from.
[event][EventData][StageName]	Name of the Send Custom Data action stage.
[event][EventData][StageType]	The type of the Send Custom Data action stage.
[event][EventData][StartDate]	The start date of the session that the Send Custom Data action is running on.
[event][EventData][ProcessName]	The name of the process that the action is being called from.
[event][EventData][PageName]	The name of the page of the process the custom data action is on.

Event	Description
[event][EventData][ObjectName]	The object that the data is coming from – will always be "Data gateways".
[event][EventData][actionName]	The action that the data is coming from – will always be "Send Custom Data".

Directing data to outputs based on content

When session logs and dashboard data are sent to separate text files, conditional statements can be applied to the outputs that will only pass events to an output if it meets one or more conditions of those conditions. This allows outputs, customized in the advanced editor or created in an external text editor, to support Logstash functionality not provided in the Data Gateway Configuration wizard. For example, outputs can be edited to only send data for specified processes or dashboard tiles.

In this example the conditional statements around the file outputs check for a certain EventType value. Session logs have an event type of 1, and dashboards have an even type of 2.

```
input {
jdbc {
jdbc_driver_library => "..\sqljdbc_4.2\enu\jre8\sqljdbc42.jar"
jdbc_connection_string =>
"jdbc:sqlserver://localhost\sqlexpress:1433;databaseName=a;"
jdbc_driver_class => "com.microsoft.sqlserver.jdbc.SQLServerDriver"
jdbc_user => "<%Data Gateways BP Database SQL User.username%>"
jdbc_password => "<%Data Gateways BP Database SQL User.password%>"
statement => "delete top(3000)from BPADataPipelineInput with (rowlock, readpast)
output deleted.eventdata"
schedule => "*/3 * * * * *"
}
}
filter {
json {
source => "eventdata"
target => "event"
}
}
output {
if [event][EventType] == 2 and [event][EventData][Source] == "Dashboard 1" {
file {
path => "C:\dashboardlogs.txt"
codec => line { format => "%{event}"}
}
}
if [event][EventType] == 1 {
file {
path => "C:\sessionlogs.txt"
codec => line { format => "%{event}"}
}
}
}
```

For information about the structure of events pulled from the Blue Prism database see Event structure.

Advanced configuration for database outputs

Database outputs, configured in the wizard, must adhere to an expected format:

- There must be an eventType column of type integer – this stores the type of the event.
- There must be an eventData column of type nvarchar(max) – this stores the events serialized to a JSON string.

In advanced configurations, the columns of the table and the data inserted into the table can be customized.

In this example, certain fields from the session log events are sent to the tableabc table in a database.

The jdbc database output inserts the session number, process name, and attributexml fields from the session log into the appropriate columns of the tableabc table.

```
input {
jdbc {
jdbc_driver_library => "..\sqljdbc_4.2\enu\jre8\sqljdbc42.jar"
jdbc_connection_string =>
"jdbc:sqlserver://localhost\sqlexpress:1433;databaseName=a;"
jdbc_driver_class => "com.microsoft.sqlserver.jdbc.SQLServerDriver"
jdbc_user => "<%Data Gateways BP Database SQL User.username%>"
jdbc_password => "<%Data Gateways BP Database SQL User.password%>"
statement => "delete top(3000)from BPADataPipelineInput with (rowlock, readpast)
output deleted.eventdata"
schedule => "*/3 * * * * *"
}
}
filter {
json {
source => "eventdata"
target => "event"
}
}
output {
if [event][EventType] == 1 {
bpjdbc {
connection_string =>
"jdbc:sqlserver://TheServer;databaseName=MyDB;integratedSecurity=true;"
driver_jar_path => "..\sqljdbc_4.2\enu\jre8\sqljdbc42.jar"
driver_class => "com.microsoft.sqlserver.jdbc.SQLServerDriver"
statement => ["insert into tableabc(EventType, EventData) values(?, ?)", "[event]
[EventType]", "[event][EventData]"]
}
}
}
```

Custom configuration examples

For a full list of all events and more information about the event structure, see Event structure.

Filter events and divert outputs using conditional statements

In this example the [event][EventType] field is used to send event types to separate files depending on whether event type is a session log (EventType == 1) or a published dashboard (EventType == 2).

The event type for custom object data (EventType == 3) is not specified and so any data of this type in the Data Gateways engine is not included in discarded.

```
input {
jdbc {
jdbc_driver_library => "..\sqljdbc_4.2\enu\jre8\sqljdbc42.jar"
jdbc_connection_string =>
"jdbc:sqlserver://localhost\sqlexpress:1433;databaseName=a;"
jdbc_driver_class => "com.microsoft.sqlserver.jdbc.SQLServerDriver"
jdbc_user => "<%Data Gateways BP Database SQL User.username%>"
jdbc_password => "<%Data Gateways BP Database SQL User.password%>"
statement => "delete top(3000)from BPADataPipelineInput with (rowlock, readpast)
output deleted.eventdata"
schedule => "*/3 * * * * *"
}
}
filter {
json {
source => "eventdata"
target => "event"
}
}
output {
if [event][EventType] == 2 and [event][EventData][Source] == "Dashboard 1" {
file {
path => "C:\dashboardlogs.txt"
codec => line { format => "%{event}"}
}
}
if [event][EventType] == 1 {
file {
path => "C:\sessionlogs.txt"
codec => line { format => "%{event}"}
}
}
}
```

Send events based on session log process names

In this example events to a particular output based on a process name from a session log. There are two outputs:

- All events get sent to the C:\allevents.txt text file
- Session log events from the *Process123* process are additionally sent to the specified HTTP endpoint.

```
input {
jdbc {
jdbc_driver_library => "..\sqljdbc_4.2\enu\jre8\sqljdbc42.jar"
jdbc_connection_string => "jdbc:sqlserver://localhost:1433;databaseName=ExampleDB;"
jdbc_driver_class => "com.microsoft.sqlserver.jdbc.SQLServerDriver"
jdbc_user => "<%SQL Serv.username%>"
jdbc_password => "<%SQL Serv.password%>"
statement => "delete top(3000)from BPADataPipelineInput with (rowlock, readpast)
output deleted.eventdata"
schedule => "*/3 * * * * *"
}
}
filter {
```

```
json {
source => "eventdata"
target => "event"
}
}
output {
if [event][EventType] == 1 {
file {
path => "c:\allevents.txt"
codec => line { format => "%{event}"}
}
}
if [event][EventType] == 1 and [event][EventData][ProcessName] == "Process123" {
bphttp {
url => "localhost:8080/api/post"
http_method => "post"
headers => {"Authorization" => "Basic <base64><%SQL Serv.username%>:<%SQL
Serv.password%></base64>"}
}
}
}
```

Credentials in custom configurations

When credentials or other sensitive data is required in the configuration, they should be added to a Blue Prism credential and then referenced in the configuration by the credential name.

When creating a Blue Prism credential for use in Data Gateways configurations, the credential type must be *Data Gateway Credential*. These credentials are accessible only by the Data Gateway system and are not accessible to Blue Prism processes.

Credentials can be referenced in the configuration using the syntax `<%{credentialname}.{property}%>`, where `{credentialname}` is the name of the credential and `{property}` is the name of the property in the credential.

For example, to use the username of a credential named *cred1* the configuration code would be `<%cred1.username%>` and `<%cred1.password%>` respectively.

Credential custom properties can be accessed using the property name.

Data Storage when endpoint unavailable

Temporary data storage

If a connection to an HTTP, Splunk, or Database endpoint can't be made when session log data is sent from the Blue Prism database to the Data Gateways engine, the data for the associated output is stored temporarily in a dedicated file within the Data Gateways engine. When the endpoint becomes available the data is resent to the specified endpoint and deleted from the file.

> This functionality will only work if the initial connection to the Data Gateways engine was made successfully. The temporary storage of data will only be triggered if a connection to the specified endpoint cannot be made during a previously established session. Data that cannot be temporarily stored is added to 'dead letter queue' storage, as described below.

The session log data is held indefinitely until the endpoint becomes available and will continue to store additional session log data as subsequent events are written to it.

'Dead letter queue' data storage

If the destination endpoint is available but an unexpected error occurs while transferring session log data to it, the log data will be removed from temporary storage and will be written to the dead letter queue along with information about the error. The dead letter queue file is stored by default in the following directory: logstash\data\queue\main on the application server that runs the Data Gateways process. The data is stored in a non-human readable format. The default maximum size for a data storage file in the Data Gateways engine is 1024 MB (1 GB). If a file reaches this limit, a new file is automatically created to store the excess data.

There is no in-built mechanism that will inspect these files and make further attempts to transfer the data to the target system. It is recommended that you implement a system to clear out this data on a regular basis to prevent a build-up of stored data.

Blue Prism output extensions

By default, the output extension BPHTTP is used for HTTP and Splunk, and BPJDBC is used for Database output types. These Blue Prism extensions are integral to the automatic storage of session log data when an endpoint is unavailable.

Upgrading from 6.5

In Blue Prism version 6.6, BPHTTP and BPJDBC output extensions replaced HTTP and JDBC. If upgrading Blue Prism from 6.5, the output types must be updated to use these output extensions to utilize the endpoint unavailable data storage functionality. This can be done by either creating new Data Gateways configuration, or manually editing the existing configuration via advanced or custom configuration and replacing HTTP with BPHTTP , and JDBC with BPJDBC .

File and custom output types

This functionality does not apply to File or custom configured output types. When using these output types, we recommend enabling Write session logs to database to avoid data loss.

Troubleshooting Data Gateways

How do I check that the Data Gateways system is functioning as expected?

There are a number of ways to check Data Gateways for issues:

- **Check the status of the Data Gateways process in control room**

  If it is in an error state, the status message will provide information about what the problem is.

- **Check the event logs on the application server**

  Any Data Gateways process errors will be output into the event log from the Logstash Process Manager source. The standard output stream from the Logstash process is also written to the event log in the event of an error, which can help identify Logstash specific errors.

- **Check the Logstash logs on the application server**

  The logs are written to C:\Logstash\logstash\logs\.

Should I use a dedicated application server for Data Gateways?

For the majority of installations, existing application servers can be used for Data Gateways. However, where Data Gateways is being used to process large sets of data, or to direct data to a number of targets, it may be appropriate to deploy Data Gateways onto dedicated Blue Prism application servers which are used only for this purpose.

How do I set up Data Gateways Splunk outputs to use HTTPS?

For details about how to achieve this, see:

https://support.blueprism.com/en/support/solutions/articles/7000076645.

What do I do if no Logstash log files are generated?

If no Logstash files are generated, ensure the user running the Data Gateways process has permission to write to the Logstash directory. Navigate to C:\Logstash and select **Properties** from the context menu. On the **Security** tab, check the user has the Write permission enabled. If this permission is not granted, exceptions may occur and logstash files will not be generated.