BeyondTrust

# Privilege Management Reporting Installation Guide 5.4.x

***Powered By Defendpoint***



TC:7/12/2019

# Table of Contents

# Introduction and Overview

This document explains how to install and configure BeyondTrust Privilege Management Reporting, which enables organizations to monitor and report on activity from Windows and Mac desktops and servers.

There are several methods available for centralizing audit data. The most common is Windows Event Forwarding. Privilege Management Reporting can use Windows Event Forwarding to centralize audit data to one or more Windows Event collector server hosts.

> For more information, please see the *Privilege Management Reporting Event Centralization* guide available from BeyondTrust.

Once audit data are collected, one (or more) instances of the BeyondTrust Event Parser component load the data into the BeyondTrust Privilege Management database on a Microsoft SQL Server instance. All audit event data are stored in one logical SQL Server instance.

Reports provide visibility to the audit data and are implemented as custom reports in Microsoft SQL Server Reporting Services 2012 or later.

Microsoft SQL Server Reporting Services is typically hosted independently from the audit events SQL Server database instance, except for small implementations and evaluation scenarios where it may share the audit database server host.

Reporting is also available in the BeyondTrust Privilege Management ePO Edition. With the Privilege Management ePO Edition, event centralization and report presentation are built on the ePO framework agent and ePO server, with audit data storage in Microsoft SQL Server as described in this guide.

> For more information, please see the *Privilege Management ePO Administration Guide* for instructions specific to the ePO Edition.

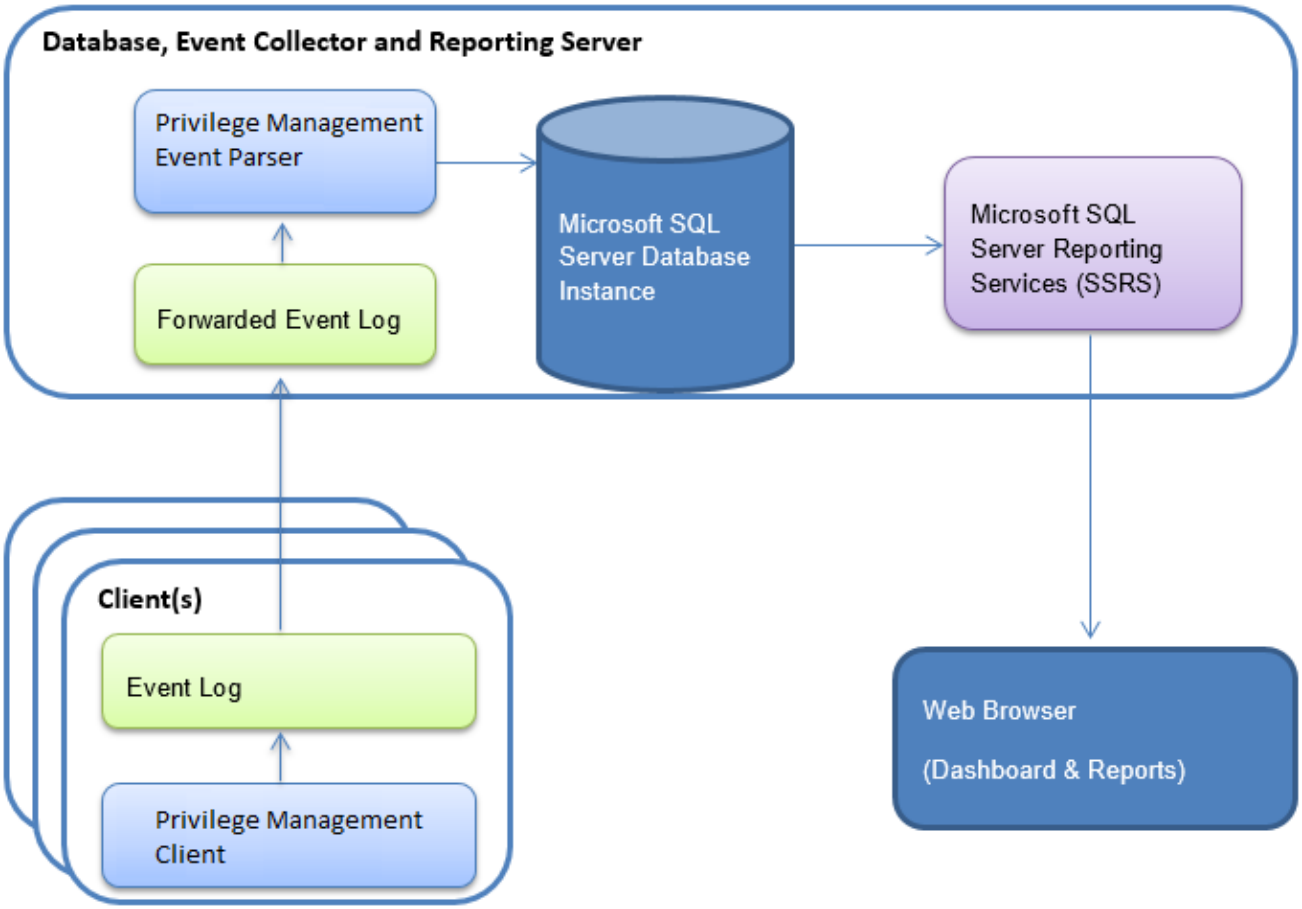
## Configuration Options

There are two options for deploying the solution:

**Option 1**: Use a single box solution, which is suitable for evaluating the product, or for SME installations.

**Option 2**: Use a scaled out deployment, which is recommended for larger production environments.

## Option 1 - Single Box Solution




In this deployment scenario, one server provides all functions.

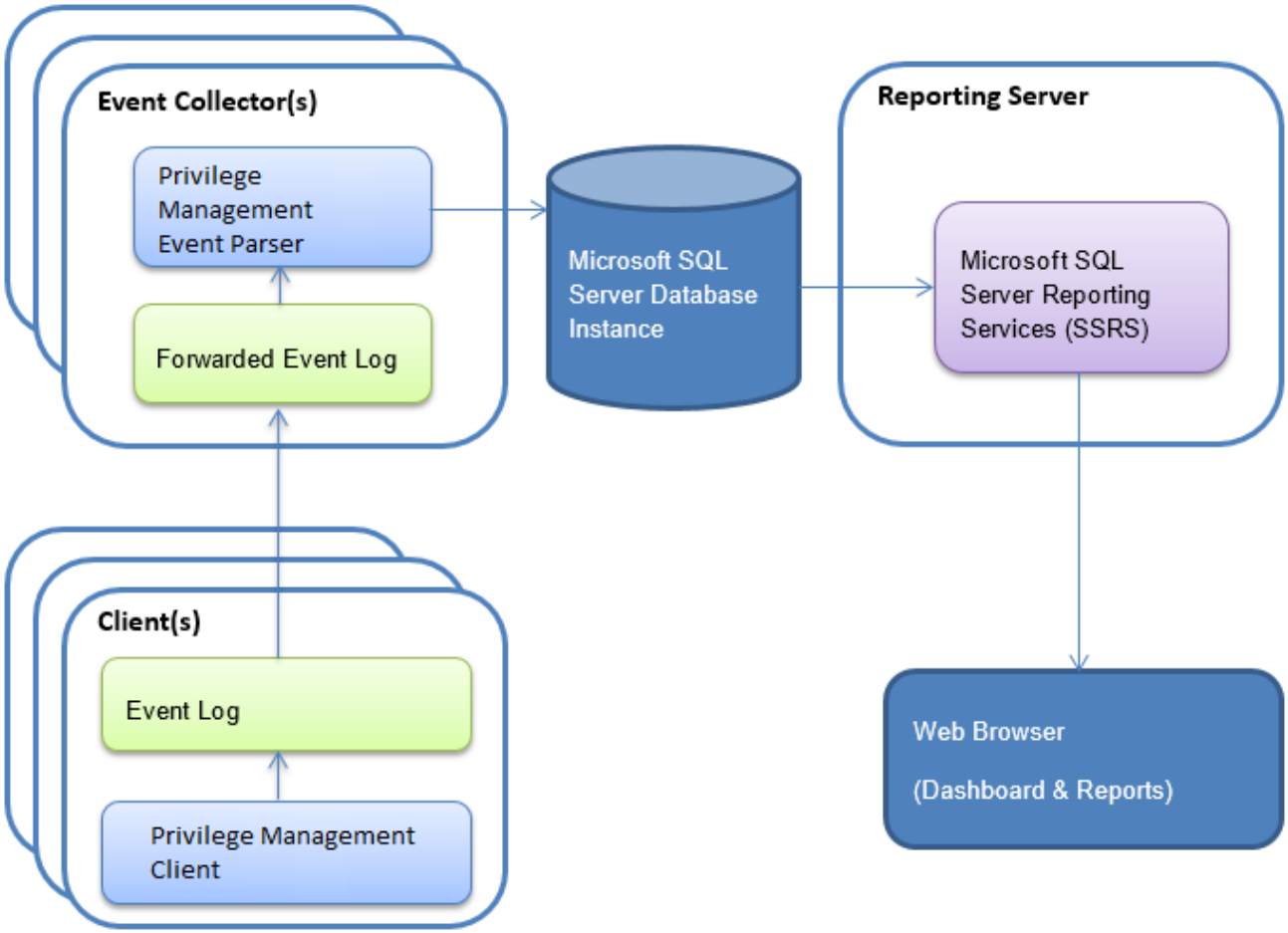
- The server must be running Windows Server 2012 or later.
- SQL Server 2012 R2 or later must be installed on theserver.

For an evaluation:

- A Windows Client, such as Windows 8, is supported.
- SQL Server Express is supported.

Select the Reporting Services feature in the feature selections page of the Microsoft SQL Server installer. To install Reporting Services, use **Native Mode**.

### Option 2 - Enterprise Scaled Out Deployment




In this deployment scenario the Event Collectors, Database and Reporting Server are installed on dedicated servers.

## SQL Server Database

The database is a repository for the data collected from the clients.

- The minimum version required is SQL Server 2012.
- Clustered databases are supported.
- When you install SQL Server, you must select a case insensitive collation. We recommend you select **Latin1_General_CI_AS**.
- iC3 installations require Azure SQL Server which is also supported.
- Windows Integrated Authentication must be used for Event Parser connections.
- SSRS connections can use either Windows Integrated Authentication or SQL Server Authentication.
- TCP/IP connections must be enabled on the SQL Server to allow the Event Collector service to submit events.
- Microsoft SQL Server CE is not supported.

The database is created during the installation of the Privilege Management Reporting Database component. By default, the database is named **BeyondTrustReporting**. The installation provides the option to provide a custom database name.

## Event Collector (Server)

The Privilege Management Event Parser is a service that detects and submits new Privilege Management Events to the database. Typically, the Event Parser is installed on a dedicated Windows Event Collector Server, and by default scans the ForwardedEvents Log for new events.

The Event Parser service can be configured to scan the Application Event Log if required, by editing the following Registry value:

```
HKEY_LOCAL_MACHINE\Software\Avecto\Privilege Guard Event Parser\
REG_SZ "EventLog"
```

The Event Collector host should be built on Windows Server. The following versions are supported: 2012, 2014, and 2016.

Ideally, the server is dedicated to this role.

You may configure multiple Event Collector servers that feed into a single database.

***Note:** The Event Parser services are the only components which establish direct connections to the events database. This keeps the number of concurrent connections to a minimum.*

## Report Server

- SQL Server Reporting Services (SSRS) 2012 or later is required.
- The server must be dedicated to this role.
- The events database and SSRS can be hosted on the same SQL Server instance.
- We recommend that the SSRS instance be separate from the database instance to prevent performance issues on the database.

The SSRS reports are installed and pre-configured during the installation of the Privilege Management Reporting Pack component. By default, the SSRS instance is named **ReportServer**. You can provide a custom name during the SQL Server installation.

## Client Configuration

Windows Event Forwarding is the technology used to gather events from the clients running Privilege Management Reporting.

For more information, please see the *BeyondTrust Event Centralization* guide for installation and configuration details on Windows Event Forwarding.

- Event forwarding must be configured for all computers running the Privilege Management Reporting Client that need to forward events.
- The minimum OS level required on each client is Windows 7.
- Events can be forwarded to any of the supported Windows Server OS versions (Windows Server 2012 or later).
- Each client requires Windows Remote Management (WRM) 1.1 or later installed.

## Database Size and Resource Consumption

## Data Retention Considerations

The Audit Event database and Microsoft SQL Server Reporting Services database that supports BeyondTrust Privilege Management Reporting can be hosted and scaled independently.

It's important to identify the length of time that Privilege Management audit event data must be retained in the Privilege Management database as it drives resource utilization projections, and initial allocation.

Privilege Management Reporting is designed to report on activity in recent time, not as a long term archival data storage solution.

- BeyondTrust provides a database purge utility that may be used to purge data manually, or automatically on a configured period to ensure database growth is capped.
- Unlimited database growth reduces query execution performance, and increases resource utilization for queries.

***Note:*** *Before purging large sets of data, please ensure your SQL Transaction logs can grow to accommodate. It may be necessary to delete data in stages when setting this up for the first time.*

To facilitate your decision making regarding retention time in the Privilege Management database, please see the following sections in our standard documentation:

- Description of the views of data exposed in Privilege Management Reporting - the *Reporting Dashboards Guide*.
- Description of the events audited by Privilege Management in the *Administration Guide - Auditing and Reporting - Events*.
- Description of the Workstyle parameters. You may consider these as the fields that are collected in the audit events, eventually stored in the Privilege Management Audit Events database. *Privilege Management Administration Guide - Workstyle Parameters*.

# Database Sizes

The Audit Event database must be sized to accommodate substantial data volume, matching the number of clients generating audit data, and the desired retention period.

Database storage requirements may be estimated roughly using the following calculation:

```
Number of hosts
X Number of events per host per day
X 5Kb per event
X Number of retention days
```

For example, an organization of 10,000 hosts, with each host generating an average of 15 events per day, requiring a 30 day retention needs a database capacity of:

```
10,000 X 15 X 5 X 30 = 22,500,000Kb, or 21.5Gb
```

A typical event volume is 10-20 events per host per day and varies based on Privilege Management Reporting auditing configuration, user job function (role/workstyle), and user activity patterns.

> For more information, please see the Privilege Management Database sizing calculator to further explore database sizing and growth expectations.

Database resource utilization (CPU, Memory) is highly variable depending on the hardware platform.

### Example Use Case Volumes

Based on an organization of 10,000 hosts requiring a 42 day (six weeks) retention.

Discovery: Between 40 – 60 events per machine per day

(4.6K per event (based on real world data))

Average total: **67.06 GB**

Production: Between 2 – 10 events per machine per day

(4.6K per event (based on real world data))

Average total: **5.66GB**

> ***Note:*** *If the number of events* ***per machine per day*** *is raised to 15, then the Average total increases to 16.99 GB.*

### Key considerations

- Volume of inbound audit event records: As noted above, the number of events per hour may be estimated following simple calculations. The audit event records are bulk inserted (no integrity checks, transactions) in batches of 100 by the Event Parser, and post-processed by a scheduled job that normalizes the audit event records into the Audit Event database schema.
- Queries triggered from Microsoft SQL Reporting Services Reports: As the database grows in size, the resource impact of the reporting platform queries becomes important.
- The volume of data maintained in the audit event database will affect the duration and resource cost of these queries.

To maintain good performance, we recommend the "Purge Privilege Management Data" on page 35 is used to limit the timespan of audit event data retained in the database.

- Finer-grained audit data management and clean-up is possible using the Privilege Management Reporting Database Administration Dashboard. The Database Administration Dashboard allows the purging of audits related to specific applications and suppression of incoming audit items related to those applications.

For more information, please see the Database Administration description in the *Reporting Dashboard Guide*.

***Note:*** *Before purging large sets of data, please ensure your SQL Transaction logs can grow to accommodate. It may be necessary to delete data in stages when setting this up for the first time.*

## SSRS Sizes

The Microsoft SQL Server Reporting Services database remains relatively small and capacity planning is not important.

A dedicated server should be sized according to Microsoft SQL Server Minimum Specifications and assessed periodically.

# Privilege Management Reporting Database Installation

Install the Privilege Management Reporting Database before the Event Parser. As part of the install, you will set the database connection details, and the installer will create the Privilege Management database if it doesn't already exist.

> ***Note:*** *The Privilege Management Reporting Database installer creates a database and database permissions through embedded SQL scripts. If your database administration team does not allow creation of databases or database permissions by installers, please contact BeyondTrust Technical Support for assistance with an alternative approach.*

## Pre-Installation Tasks

### Accounts

Before starting with the installation of the Privilege Management Reporting Database, we recommend the following accounts are created.

#### Accounts Required for Installation

| Name | Details | Account Type | Permissions / Rights |
|---|---|---|---|
| DatabaseCreator | Used by the Reporting Database installer to create the Privilege Management database | Windows account or SQL Authentication account | SQL Server permission – sysadmin<br>The database must be installed by a user whose default schema is DBO.<br>For more information, please see https://technet.microsoft.com/en-us/library/ms190387(v=sql.105).aspx<br>BeyondTrust Technical Support can assist with a manual setup in scenarios where sysadmin permissions are not permitted. |
| EventParser | Used by the Event Parser service to connect to the BeyondTrust database and write event data | Windows account | SQL Server permission - database write access Windows group members - Event Log Readers<br>Windows permission - Network access (for remote SQL Server instance) |

| Name | Details | Account Type | Permissions / Rights |
|---|---|---|---|
| ReportReader | Used by the Reporting Pack reports to allow read access to the Privilege Management database | Windows account or SQL Authentication account | Requires Log On Locally rights on server hosting SSRS<br>SELECT and EXECUTE permissions are assigned during the installation process |
| DataAdmin | Used by the Reporting Pack reports to allow write access to the Privilege Management database to purge undesired data.<br>This account and product feature is optional - please see section 4.2 Installation for more information. | Windows account or SQL Authentication account | Requires Log On Locally rights on server hosting SSRS<br>SELECT and EXECUTE permissions are assigned during the installation process |

***Note:*** *If you are using a single server, as in Deployment Option 1, then you may want to run the Privilege Management Event Parser services as the SYSTEM account. In this scenario, you can use the Database installer to configure database access for the SYSTEM account.*

***Note:*** *If Windows Authentication is selected for the SQL connection, then the account of the installing user MUST have Alter Any Login and Create Any Database permissions on the SQL Server instance for the Reporting Services instance User to be created. If you receive an error 15247, verify these permissions are granted.*

## Installation

To install Privilege Management Reporting Database, run the appropriate installation package with an account that has **Database Creator** privileges:

If you are running the installer on the database machine use:

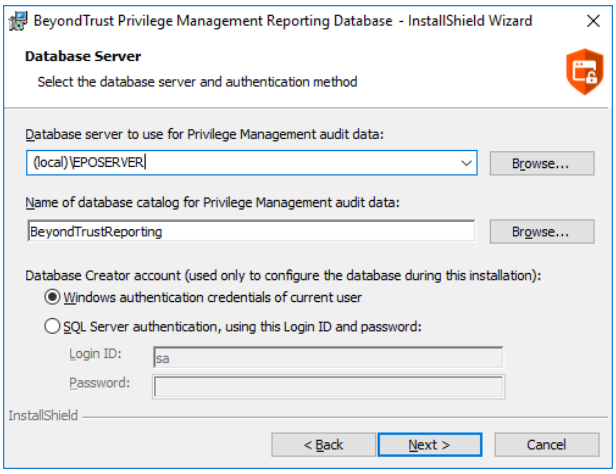
```
PrivilegeManagementReportingDatabase.msi
```

If you are running the installer on a client machine use:

```
PrivilegeManagementReportingDatabase.exe
```

1. Run the appropriate installation package and click **Next** to continue. The **License Agreement** dialog box is displayed.
2. After reading the license agreement, select **I accept the terms in the license agreement** and click **Next** to continue. The **Database server** dialog box is displayed.

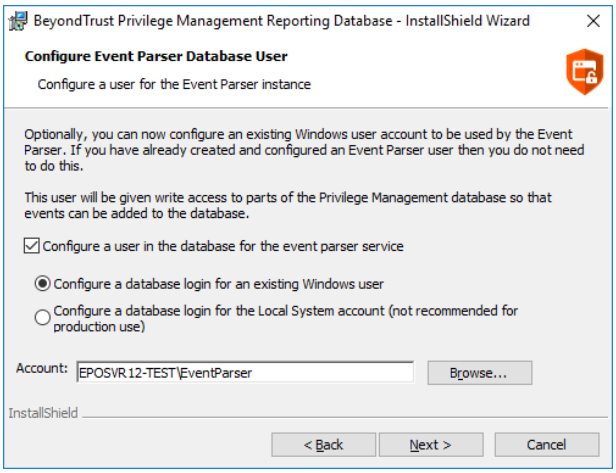
3. Enter the name of the database catalog for Privilege Management audit data. You can choose to use the current Windows user for the Database Creator user or enter credentials for a SQL account. Click **Next** to continue.



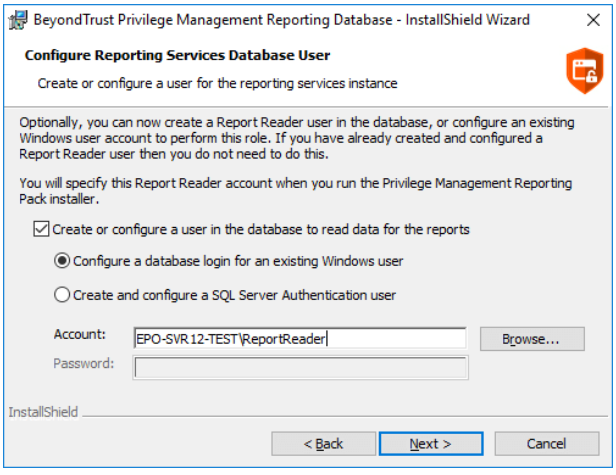

> We recommend you leave caching enabled. For more information, please see "Database Cache" on page 30.

5. The Configure Report Data Caching dialog box is displayed. Click **Next**. The **Configure Event Parser Database User** dialog box is displayed.

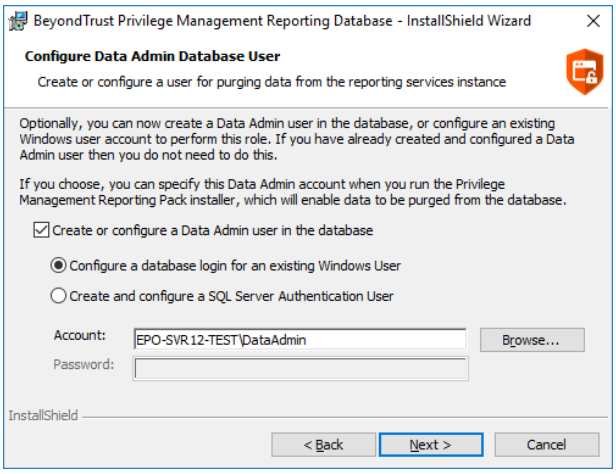
6. You must configure an Event Parser user to ensure the appropriate permissions are added for the database. You can choose to use the current Windows user for the Event Parser user or create a SQL Server account. Click **Next** to continue.




7. The **Configure Reporting Services Database User** is displayed. You must configure a Report Reader user to ensure the appropriate permissions are added for the database. You can choose to use the current Windows user for the Report Reader user or create a SQL account. Click **Next** to continue.



8. The **Configure Data Admin Database User** dialog box is displayed. You must configure a Data Admin user to ensure the appropriate permissions are added for the database. You can choose to use the current Windows user for the Data Admin user or create a SQL account. Click **Next** to continue.
9. The **Ready to Install the Program** dialog box is displayed. Click **Install**, and then click **Finish**.



# Event Parser Installation

# Pre-Installation Tasks

## Accounts

Before starting the Event Parser installation, it is recommended that the following accounts are created. The installation steps in subsequent sections of this guide will refer to these accounts.

### Accounts Required for Installation

| Name | Details | Account Type | Permissions / Rights |
|---|---|---|---|
| ERInstaller | Use this account to install the Event Parser | Windows account | Windows permission - Local Administrator |

### Accounts Required for Runtime

| Name | Details | Account Type | Permissions / Rights |
|---|---|---|---|
| EventParser | Used by the Event Parser service to connect to the BeyondTrust database and write event data | Windows Account | SQL Server permission - Database write access<br>Windows group member - Event Log Readers<br>Windows permission - Network access (for remote SQL Server instance) |

***Note:*** *If you are using a single server, as in Deployment Option 1, then you may want to run the Privilege Management Event Collector service as the SYSTEM account. In this case, you can specify the SYSTEM account as part of the installation.*

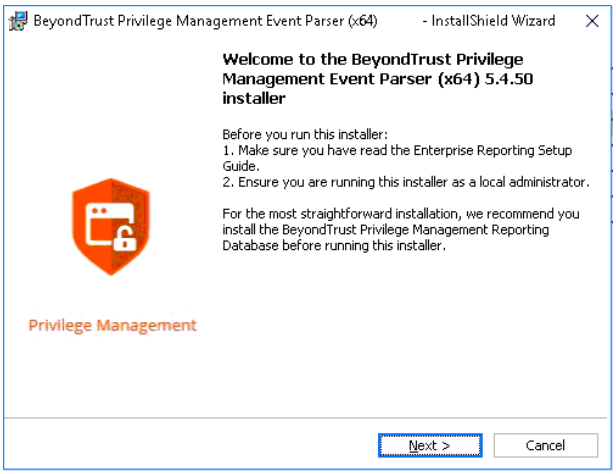
***Note:*** *The SQL Server configuration must have TCP/IP communications enabled to allow the Event Parser Service to submit events to the database.*

# Event Parser Installation

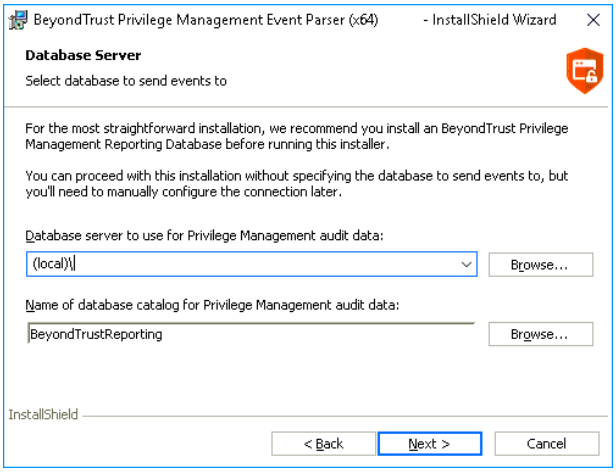
To install Privilege Management Event Parser, run the installation package with an account that has Installer privileges:

Systems must be 64-bit. Run **PrivilegeManagementEventParser_x64.exe**

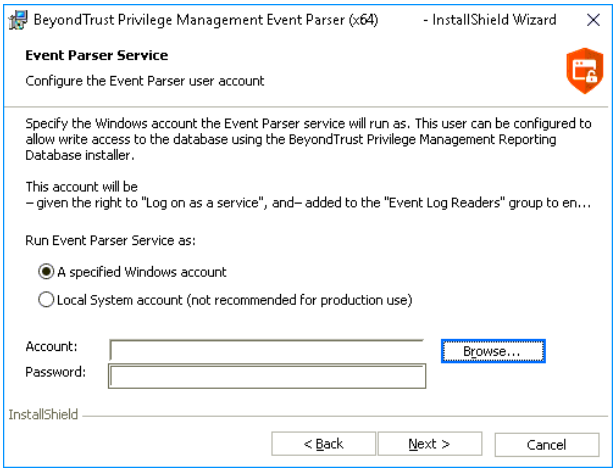
1. Run the installation package.
2. Click **Next** to continue. The **License Agreement** dialog box is displayed.
3. After reading the license agreement, select **I accept the terms in the license agreement** and click **Next** to continue. The **Destination Folder** dialog box is displayed.
4. To change the default installation directory click **Change** and select a different installation directory.




5. Click **Next** to continue. The **Database Server** dialog box is displayed.




6. Enter the details of the database server.
7. Click **Next** to continue. The **Event Parser Service** dialog box will appear.
8. Select the **EventParser** account for the **Event Parser Service**. Click the **Browse** button to select the account if desired.




> ***Note:*** *This account will be added to the Event Log Readers group on the Event Collector server. It will also be granted the Log on as a service right on the Event Collector server.*

9. Click **Next** to continue. The **Ready to Install the Program** dialog box is displayed.
10. Click **Install** to complete the installation. The **Install Shield Wizard completed** dialog box is displayed.

# Privilege Management Reporting Pack Installation

Install the Reporting Pack on the SQL Server Reporting Services instance (or the single server if using a 'single box' solution).

## Pre-Installation Tasks

## Accounts

Before starting the installation of the Reporting Pack components, it is recommended that the following accounts are created. The installation steps in subsequent sections of this guide will refer to these accounts.

### Accounts Required for Installation

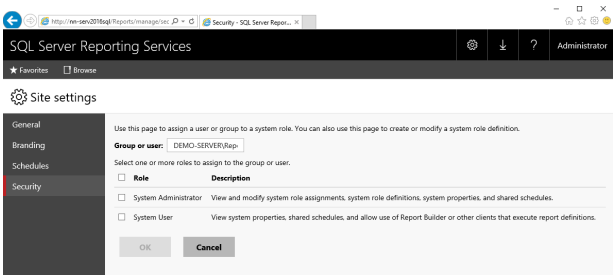
| Name | Details | Account Type | Permissions / Rights |
|---|---|---|---|
| ReportWriter | Use this account to install the Reporting Pack | Windows account or SQL Authentication account | Windows permission - Local Administrator<br>SSRS site level role - System Administrator |
| ReportReader | Used by the Reporting Pack reports to allow read access to the Privilege Management database | Windows account or SQL Authentication account | Requires Log On Locally rights on server hosting SSRS<br>SELECT and EXECUTE permissions are assigned during the installation process |
| DataAdmin | Used by the Reporting Pack reports to allow write access to the Privilege Management database to purge undesired data.<br>This account and product feature is optional - please see section 4.2 Installation for more information. | Windows account or SQL Authentication account | Requires Log On Locally rights on server hosting SSRS<br>SELECT and EXECUTE permissions are assigned during the installation process |

### Create the ReportWriter Account

To add a System Administrator role to the Reporting Services site:

1. Browse to the SQL Server Reporting Services **Report Manager** URL. The URL is located in the Reporting Services Configuration Manager, under **Report Manager URL**.
2. Click on **Site Settings**, and then select **Security**.
3. Click **Add group or user**, and enter the **DOMAIN\Username** of an authorized account.
4. Select the **System Administrator** check box.

5. Click **OK**.

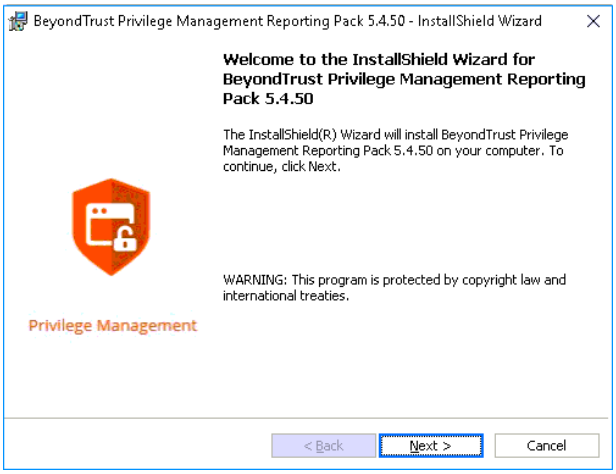



## Installation

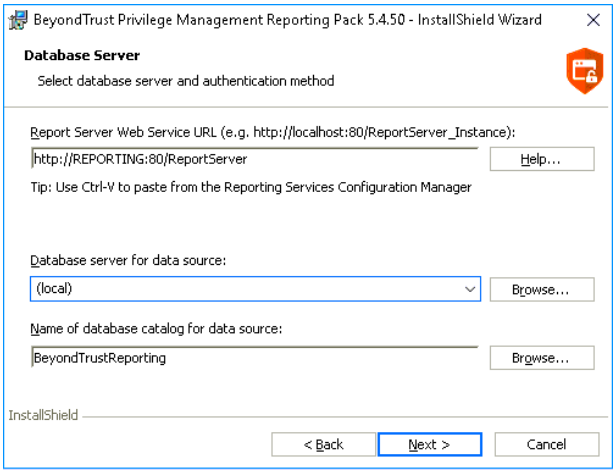
To install the Privilege Management Reporting Pack:

1. Run the **PrivilegeManagementReportingPack.exe** installation package as a user with the Report Writer permissions.

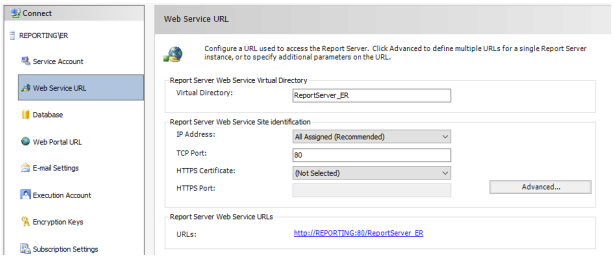
For more information, see "Pre-Installation Tasks" on page 18.



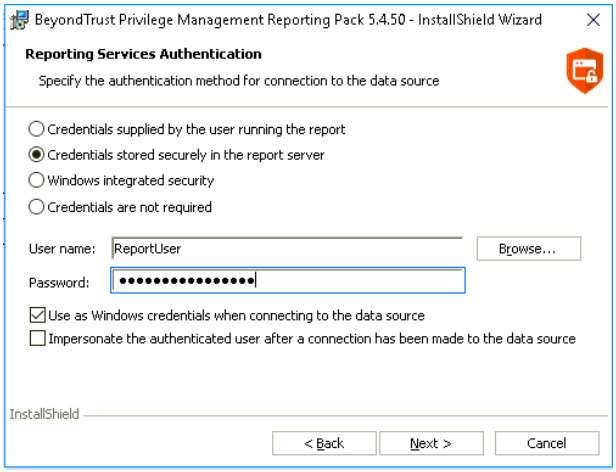

2. Click **Next** to continue. The **License Agreement** dialog box is displayed.
3. After reading the license agreement, select **I accept the terms in the license agreement** and click **Next** to continue.
4. Enter your name and the name of your organization and click **Next** to continue. The **Database Server** dialog box is displayed.




5. Enter the report server URL (the reports will fail to upload if you enter an incorrect URL). Enter the database to use by the SQL Server Reporting Services instance.




6. If you are unsure of the correct Report Server URL to use, you can find it in the **Reporting Services Configuration Manager** under **Web Service URL**:




7. Click **Next** to continue. The **Reporting Services Authentication** dialog box appears.
8. Enter the **ReportReader** account as the account used to connect to the data source.

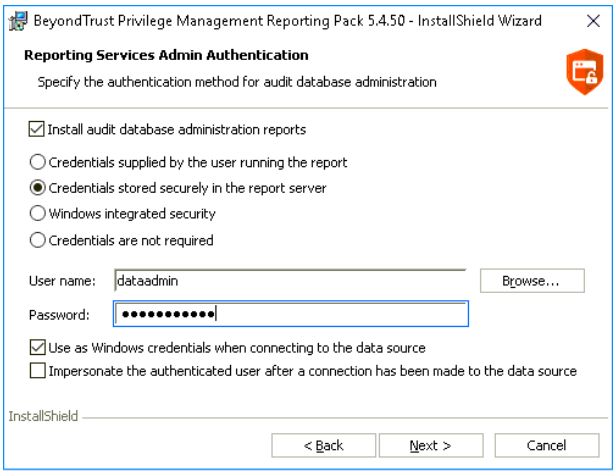
***Note:** The **ReportReader** account is used by the Reporting Services to connect to the database instance when generating dashboards and reports. The account must be the same account that was entered during the Privilege Management Database Installer.*

***Note:** If **Credentials stored locally in the report server** is not selected, then any users authorized to access Privilege Management Reporting must have their account credentials added to the **SsrsRole** database role.*

9. Click **Next** to continue. The **Reporting Services Admin Authentication** dialog box appears.
   This feature is optional and may not be desirable in environments that need tight control over purging of audit data.
10. The purpose of this report is to allow the purging (and subsequent exclusion) of applications from populating the database with unwanted data.

For more information, please see the Privilege Management Reporting Dashboard Guide.

11. Use the **DataAdmin** account for this purpose.



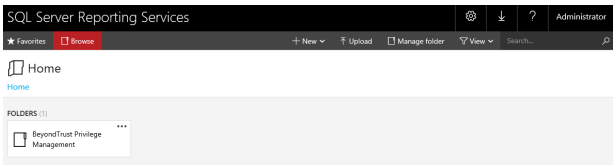

12. Click **Next** to continue. The **Ready to Install the Program** dialog box is displayed.
13. Click **Install** to complete the installation.

# Security Configuration

## SQL Server Reporting Services

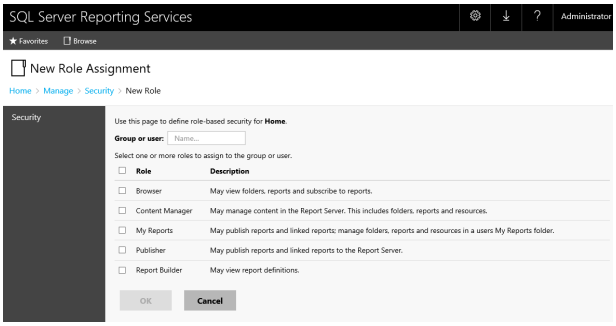
If you chose **Credentials supplied by the user running the report** or **Windows Integrated Security** for **Reporting Services Authentication**, then each user or group of users who are permitted to view reports must be granted Browse permissions in SQL Server Reporting Services (SSRS).

1. Browse to the SSRS Report Manager, using the **ReportWriter** account (you can locate the correct URL in the **Reporting Services Configuration Manager**, under **Report Manager URL**):




> ***Note:*** *You may need to run Internet Explorer with Administrator rights to initially configure the security.*

2. Click **Manage Folder** to view the security of the top level, and then click **Add group or user** to grant access to a user or group.

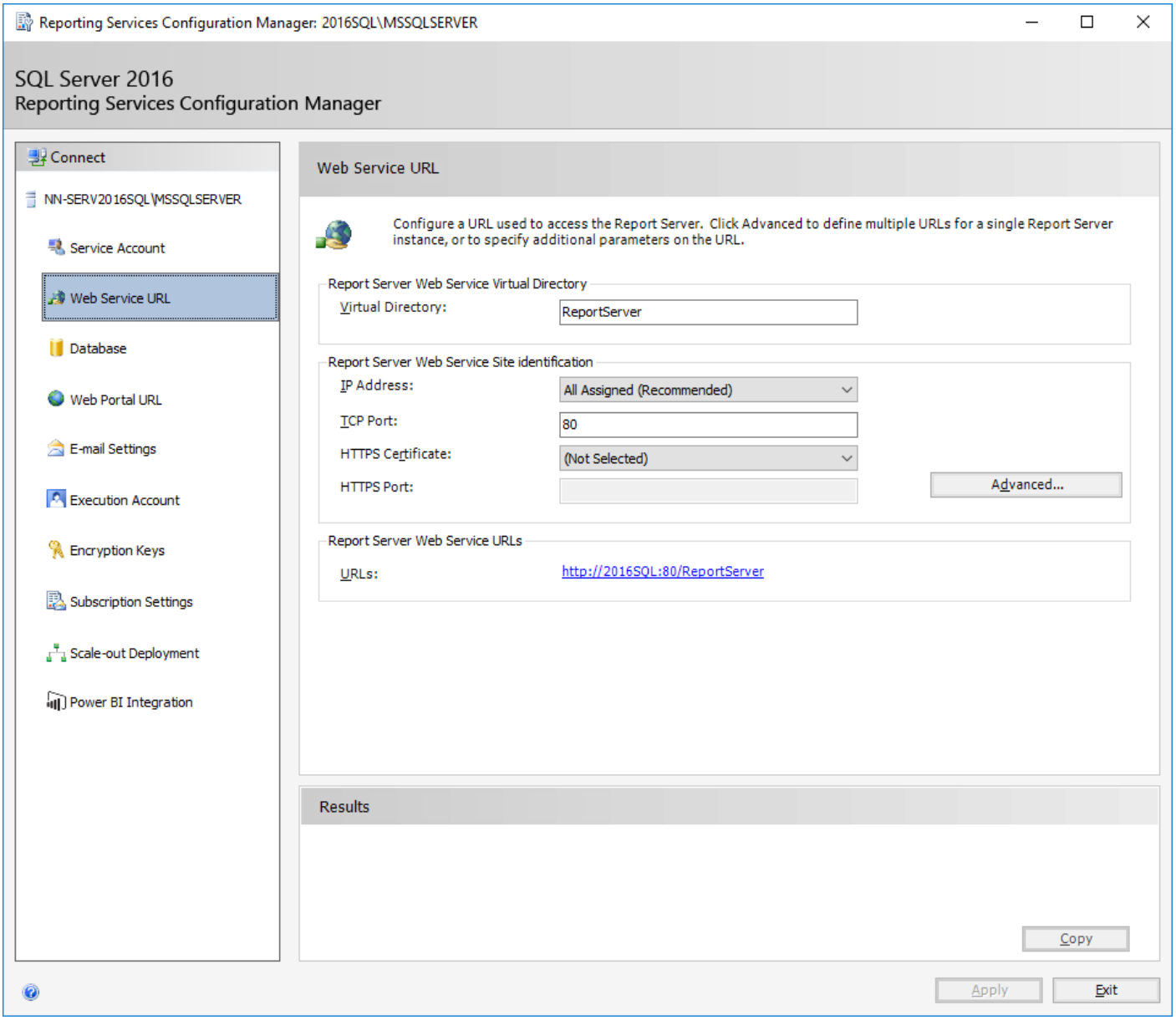



3. Enter a group or user name, select the **Browser** role, and click **OK**.
4. If the Data Administration reports were installed, security on the subfolder must restrict access to the SSRS System administrator and users authorized to purge data.

# View Dashboards and Reports

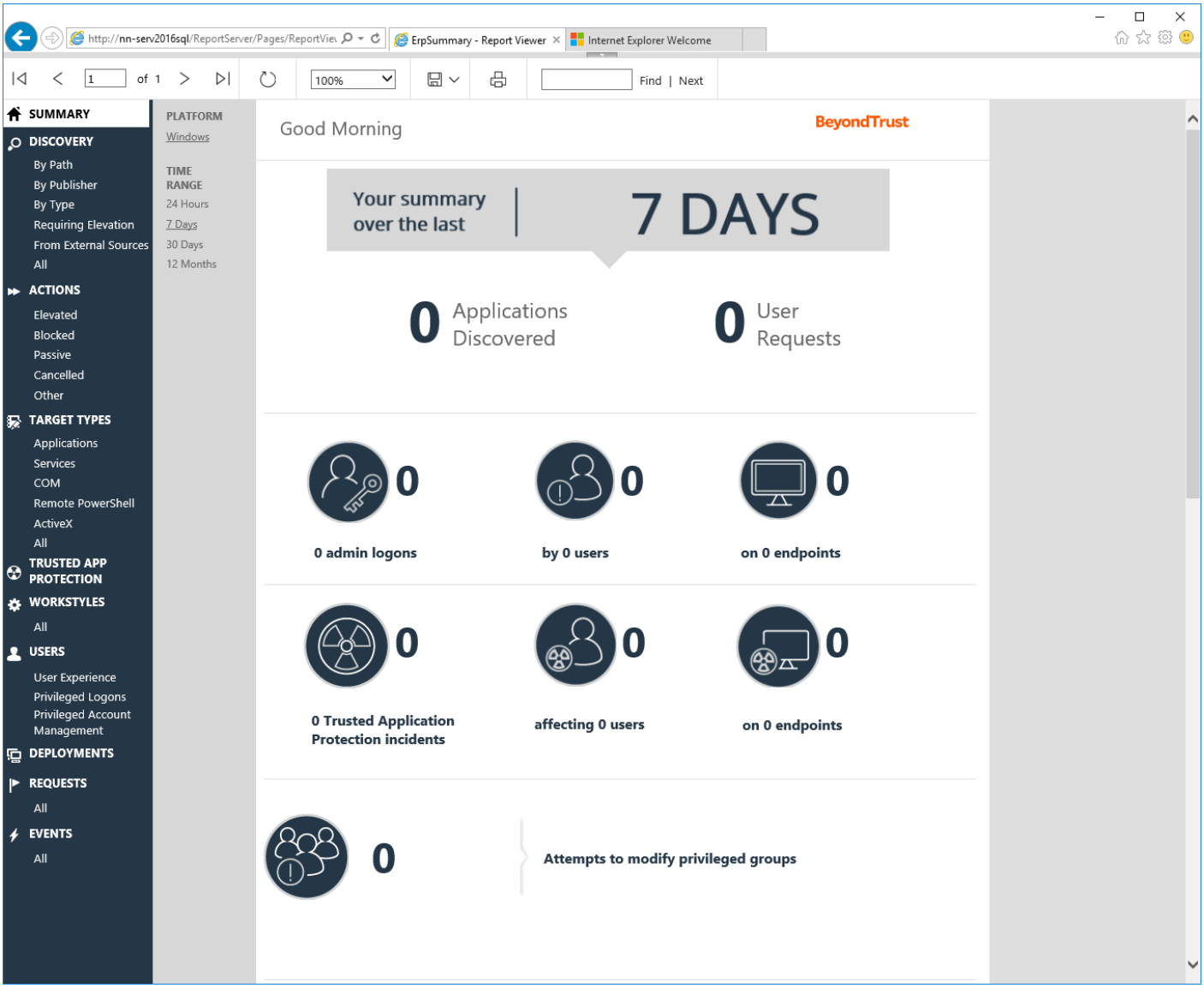
After all components are installed, and the security is configured, the dashboards and reports can be viewed. The starting point for the reports is **ErpSummary**, which you will find in the BeyondTrust Privilege Management (Privilege Guard) reports folder.

If you are unsure of the correct URL to use, you can find it in the **Reporting Services Configuration Manager** under **Web Service URL**:




***Note:*** *By default, the web service URL is https://<reportingserver>/ReportServer.*

After you navigate to the correct URL, click the BeyondTrust Privilege Management web directory, and then click the **ERP summary** report.




***Tip:** After you navigate to the **ERP Summary** report, save the address to your browser's Favorites list.*

There are several dashboards in Privilege Management Reporting:

| | |
|---|---|
| **Summary** | This report provides a summary overview of information that is available to query. |
| **Discovery** | A collection of reports that display the applications that are new to the database. The information can be used to inform workstyle updates. |
| **Actions** | Summarizes audited items categorized by the type of action taken. This allows focusing on the topic of interest. For example, elevation, blocking, etc. |

| | |
|---|---|
| **Target Types** | Summarizes specific application types that have launched and have been audited. This dashboard includes a sub-report **All**, where all raw application type data can be viewed in a tabular report. |
| **Workstyles** | Summarizes all Privilege Management workstyle usage, including coverage statistics. It identifies the top ten workstyles responsible for various application outcomes e.g. elevate, blocked, passive audited or allocated a custom token. This dashboard includes a sub report All, where all raw workstyle data can be viewed in a tabular report. |
| **Users** | Summarizes how users have interacted with messages, challenge / response dialog boxes and the shell integration within the specified time range. |
| **Deployments** | Summarizes Privilege Management Client deployments. The report shows which versions of Privilege Management are currently installed across the organization. It includes asset information about endpoints such as operating system and default language to assist with workstyle targeting. |
| **Requests** | Summarizes information about the requests that have been raised over the time frame. A blocked message with a reason entered or a canceled challenge / response message is a request. |
| **Events** | Summarizes information about the types of events raised over the time frame. It also shows how long it is since the hosts raised an event. |
| **Database Administration** | Exposes applications creating excess data that floods the database and impacts performance. It allows purging and suppression of application audits when applications are observed to create undesired audits. |

***Note:** The **Database Administration** dashboard is not available from the Privilege Management Reporting interface.*

Reports are available from the root directory if selected during the Reporting Pack installation:

https://<reportingserver>/ReportServer/

For more information, please see "Installation" on page 19.

1. Navigate to the root directory in the internet browser address bar and click **BeyondTrust Privilege Management**.
2. Click **Admin** and then click **ErpEventsAdmin**.
3. The **Database Administration** dashboard is displayed.

For more information on the purging and suppression options available from the Database Administration dashboard, please see the *Privilege Management Reporting Dashboard Guide*.

There are several summary reports for key items common throughout the Dashboards in Privilege Management Reporting. Summary Reports are available anywhere the Information logo *i* is displayed, by clicking the logo:

- **Application Summary Report**: Detailed statistical overview of a unique application.
- **Event Summary Report** : Detailed Event Log style summary of an event instance.
- **User Summary Report**: Detailed statistical overview of a user account.
- **Host Summary Report**: Detailed statistical overview of a host computer.
- **Workstyle Summary Report**: Detailed statistical overview of a Privilege Management workstyle.

Many charts and tables in the dashboards can be clicked to drill down into more detailed information.

For more information on the dashboards and reports in Privilege Management Reporting, please see the *Privilege Management Reporting Reference Guide*.

# Upgrade Privilege Management Reporting

## Assumptions

This guide assumes there is a working installation of the Privilege Management Reporting 4.0 or later installed.

Since the release of Privilege Management Reporting 4.1, the EventManagement installer is now two installers:

- PrivilegeManagementEventParser
- PrivilegeManagementReportingDatabase

In some configurations, the Event Parser is not required.

## Upgrade

The installers for the Privilege Management Database and the Event Parser must be used to manage the upgrade for on-premise databases.

> If you need to upgrade a Privilege Management database in the cloud or where you cannot use the installer, please see "Manual Upgrade" on page 29 to upgrade the database manually.

Please use the following process to upgrade the Privilege Management database and event parser:

1. Stop the **BeyondTrust Privilege Management Event Parser** service. You need to check that all events are finished processing.

Query the following tables first to check that they are empty:

- dbo.Staging
- dbo.Staging_ServiceStart
- dbo.Staging_ServiceStop
- dbo.Staging_UserLogon

Subsequently, query the following tables:

- dbo.StagingTemp
- dbo.StagingTemp_ServiceStart
- dbo.StagingTemp_ServiceStop
- dbo.StagingTemp_UserLogon

All remaining events are processed after the tables are empty.

2. Stop the **SQL Server** service.
3. Uninstall the Privilege Management Reporting Pack.
4. Restart the **SQL Server** service.
5. Load SQL Server Reporting Configuration Manager and connect to your database. Navigate to the Reporting link and use the drop-down to delete the top level folder.

6. Run the Privilege Management Database installer to upgrade the database. Ensure you point the installer to your existing Database server and Privilege Management database name when prompted.

> ***Note:*** *If you installed Privilege Management Reporting from version 5.1 (or later), the default name for the database is* ***BeyondTrustReporting****. If you installed a previous version, the default name is* ***AvectoPrivilegeGuard****. Alternatively, you may have chosen a different database name.*

> ***Note:*** *If you see an error message that states "Please stop CopyFromStaging from running before upgrading the database", then ensure no new events are processing by querying the above tables and try again.*

6. Run the Privilege Management Reporting pack to upgrade the reports. Ensure you point the installer to your existing Database server and Privilege Management database name when prompted.
7. Upgrade the BeyondTrust Privilege Management Event Parser. Ensure you point the installer to your existing Database server and Privilege Management database name when prompted.

This upgrade path can be applied to both standalone Privilege Management configurations and to configurations deployed to multiple machines.

> ***Note:*** *When you are installing Privilege Management Reporting, the Reporting Pack, the Database, and Event Parser installers should be the same version. However, you can use a different version of the Privilege Management Console and Client with Privilege Management Reporting. The Privilege Management Client generates the data that populates the reporting database. If any new features are added to the reporting pack, the pack is only populated if the Privilege Management Client is on a version that supports the data generation.*

## Manual Upgrade

Use these instructions to upgrade the Privilege Management database where you cannot use the installer or need to do a manual installation (for example, iC3 in Azure). SQL scripts are provided to manage these upgrades.

To upgrade a Privilege Management database using SQL scripts:

1. The SQL scripts are provided as part of the Privilege Management installers. Alternatively you can contact BeyondTrust Technical Support.

> ***Note:*** *There is a README file provided in this directory to assist you.*

2. Run the following SQL query to return the version of the database. For example, **4.3.16**:

```
select * from DatabaseVersion
```

> ***Note:*** *This SQL will work for Privilege Management Reporting databases 4.5 and later.*

3. Execute the upgrade script where the name is the next version number and carry on applying these until the desired version is reached.

For example, if your current database version is **4.3.16** and you want to upgrade to version **5.0.0**, run the following scripts in order:

- **Script_4.5.0_Updates.sql**
- **Script_5.0.0_Updates.sql**

Please check the SQL log for any errors and contact BeyondTrust Technical Support if necessary.

# Database Maintenance

Microsoft SQL Server Management Studio offers database administrators (DBA) a set of tools for ensuring databases are optimized, regularly backed up, and free of inconsistencies. Tasks can be created as Maintenance Plans. The SQL Server Management Studio includes a wizard for creating core Maintenance Plans.

**IMPORTANT!**

*We recommend database maintenance tasks are performed only by a qualified DBA.*

Below are a subset of Maintenance Plan tasks BeyondTrust recommends for the Privilege Management Reporting database.

Other Maintenance Plan tasks are offered by Microsoft SQL Server. Your DBA may suggest other plans that best suit your organization.

## Database Cache

When Privilege Management Reporting is deployed for larger corporate networks the amount of data can affect the performance of the reports. Reports are less interactive and load times can increase.

Upgrades to the processor, disk, or memory only make a negligible difference to the load times of reports once the database gets to a certain size. BeyondTrust has developed a solution to improve the user experience. This new feature introduces tables that store summary data to help return the information quickly.

### Data Summarization

The data summarization is linked to the current mechanism that normalizes data from the staging area. This is called by either the SQL Agent or Service Broker. Service Broker is installed by default but you can use the script **Create_ER_Database_Agent.sql** to use the SQL Agent.

The data is stored against the following intervals:

- 12 Months
- 6 Months
- 30 Days
- 7 Days
- 24 Hours

Each interval has an associated configuration that determines the amount of time until the data needs to be refreshed.

The data is refreshed depending on the interval:

- Once a day for 12-month, 6-month, and 30 day intervals
- Once an hour for 7-day interval
- Refreshed with new data for a 24-hour interval

Summarizing the grouping data adds overhead to the CopyFromStaging process. Summarizing is not done for every iteration on large installations.

The 24-hour interval is an exception; all data is refreshed since only cached data is stored for the Discovery dashboard.

## Enable and Disable Report Caching

The cache feature can be enabled or disabled when you install the Privilege Management reporting database. After installation, change cache settings using one of the following ways:

- In SQL Server Management Studio, connect to the database and run the following query:

```
UPDATE Config
SET BitValue = 1
WHERE ConfigId = 'CacheEnabled'
```

- Open SQL Server Reporting Studio and navigate to the **Admin** folder. Open the **ErpCacheAdmin** report and click **Enable** or **Disable** depending on the current state.


| INTERVAL | REFRESH TIME | CACHE EXPIRES (MINUTES) |
|---|---|---|
| 12 Months | 5/23/2019 5:00:00 AM | 1124 |
| 6 Months | 5/23/2019 5:00:00 AM | 1124 |
| 30 Days | 5/23/2019 5:00:00 AM | 1124 |
| 7 Days | 5/22/2019 11:05:00 AM | 49 |

The grouping data is cached in a table and refreshed at a configurable interval. This can be every **x** minutes or at a specific time of the day. The values that determine this behavior are in the **dbo.Config** table.

The values can be manually edited by running the following script in SQL Server Management Studio.

```
DECLARE @Interval NVARCHAR(40)= NULL -- Set the interval Last12Months, Last6Months, Last30Days,
Last7Days
Set the local time of the server to run fresh cache
-- daily for 12 / 6 Month, 30 Day eg '13:00'
-- Time on the hour for 7 Day eg '13:05'
DECLARE @LocalRefreshTime TIME = NULL
DECLARE @ForceCache BIT = NULL -- Reset the force cache
DECLARE @MyRefreshTime DATETIME
IF @LocalRefreshTime IS NOT NULL AND @Interval IS NOT NULL
BEGIN
DECLARE @UTCOffset INT = DATEDIFF(MINUTE, GETDATE(), GETUTCDATE())
DECLARE @MinutesPastMidnight INT = DATEDIFF(MINUTE, '00:00:00.000', @LocalRefreshTime)
SET @MyRefreshTime = DATEADD(MINUTE, @MinutesPastMidnight + @UTCOffset, CAST(CAST(GETUTCDATE() AS
DATE) AS DATETIME))
UPDATE [dbo].[Config]
SET DateTimeValue = ISNULL(@MyRefreshTime, DateTimeValue)
WHERE ConfigId = 'Cache' + REPLACE(REPLACE(@Interval, 'Last', ''), ' ', '') + 'Refresh'
END
IF @ForceCache IS NOT NULL
UPDATE [dbo].[Config]
SET BigValue = @ForceCache,
DateTimeValue = GETUTCDATE()
WHERE ConfigId = 'ForceCache'
```

## Rebuild Indexes on SQL Server

Microsoft SQL Server maintains indexes whenever events and data are added to or purged from the Privilege Management database. Over time, these indexes can become fragmented. Heavily fragmented indexes can cause degradations in database performance and result in increased report generation times. Rebuilding the indexes will improve database performance and restore report generation times to their optimum.

Therefore, we recommend a maintenance plan is applied to SQL Server to ensure that indexes are regularly rebuilt.

If you are not sure Privilege Management Reporting database indexes are causing a degradation in performance, then you can observe their percentage fragmentation in SQL Server Management Studio. The Processes table of the Privilege Management database is typically a high volume table, and most likely will become fragmented first.

To check the index fragmentation of the Privilege Management database in SQL Server Management Studio:

1. Navigate to **Databases > BeyondTrust Privilege Management(Privilege Guard) > Tables > dbo.Processes > Indexes**.
2. Right-click on **IDC Processes** and select **Properties**.
3. In the **Index Properties** dialog box, select **Fragmentation**.
4. The **Total Fragmentation** is displayed as a percentage.

***Note:** If you want to manually rebuild indexes for a table, you can right-click the **Indexes** node of any table and select **Rebuild All**.*

## Rebuild Indexes SQL Azure

As indexes grow they are not stored contiguously on disk and the database becomes fragmented. This results in degraded performance as more disk reads are required to load an index into memory.

It is best practice to regularly rebuild indexes. This can have a significant performance benefit.

Azure does not support rebuilding indexes using SQL Server Management Studio. It also does not support maintenance plans and does not use a SQL Server Agent. Therefore, a manual approach to rebuilding indexes is required.

## Check for Index Fragmentation

The following SQL query returns a list of the indexes in the database, with the most fragmented first.

```
SELECT
OBJECT_SCHEMA_NAME(ips.OBJECT_ID) 'Schema',
OBJECT_NAME(ips.OBJECT_ID) 'Table',
i.NAME,
ips.index_id,
index_type_desc,
avg_fragmentation_in_percent,
avg_page_space_used_in_percent,
page_count
FROM
sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, 'SAMPLED') ips
INNER JOIN
sys.indexes i
ON (ips.object_id = i.object_id)
AND
(
ips.index_id = i.index_id
)
ORDER BY
avg_fragmentation_in_percent DESC
```

A sample output of this query is shown below:

Results | Messages

| | Schema | Table | NAME | index_id | index_type_desc | avg_fragmentation_in_percent | avg_page_space_used_in_percent | page_count |
|---|---|---|---|---|---|---|---|---|
| 1 | dbo | DistinctApplications | PK_DistinctAppID | 1 | CLUSTERED INDEX | 99.2906938547078 | 60.2151964418088 | 82193 |
| 2 | dbo | Events | IDX_Events1 | 6 | NONCLUSTERED INDEX | 99.2466983287115 | 44.8708920187794 | 94119 |
| 3 | dbo | Processes | IDX_Processes5 | 19 | NONCLUSTERED INDEX | 99.2323369565217 | 45.1562268346924 | 117760 |
| 4 | dbo | Processes | UK_Processes_ProcessGUID | 23 | NONCLUSTERED INDEX | 99.231222882177 | 45.3426241660489 | 94956 |
| 5 | dbo | Events | PK_Events | 2 | NONCLUSTERED INDEX | 99.1557513693737 | 44.7456510995799 | 83980 |
| 6 | dbo | Applications | IDX_Applications1 | 2 | NONCLUSTERED INDEX | 99.1266375545851 | 67.4136397331357 | 687 |
| 7 | dbo | Domains | IDX_Domains1 | 2 | NONCLUSTERED INDEX | 99.1176470588235 | 67.6642204101804 | 340 |
| 8 | dbo | ApplicationInstallations | PK_ApplicationInstallations | 1 | CLUSTERED INDEX | 99.0002954791687 | 66.1618235730171 | 20306 |
| 9 | dbo | Users | IDX_Users1 | 2 | NONCLUSTERED INDEX | 98.7577639751553 | 65.1816407215221 | 644 |
| 10 | dbo | Domains | PK_Domains | 1 | CLUSTERED INDEX | 98.7482614742698 | 61.0140721522115 | 719 |

## Rebuild Indexes

The guidance from Microsoft is that indexes be rebuilt if the fragmentation is over 30% and reorganized if the fragmentation is between 5 and 30%. Reorganizing an index is a faster, lightweight version of rebuilding and the indexes remain online.

If you want to rebuild an index the syntax is below:

```
ALTER INDEX IDX_Processes5 ON Processes REBUILD WITH (ONLINE = ON)
```

If the index can be unavailable for a short time you can speed up rebuilding time by specifying:

```
(ONLINE = OFF)
```

The syntax to reorganize an index is:

```
ALTER INDEX IDX_Processes5 ON Processes REORGANIZE
```

## Rebuild All Database Indexes

Manually rebuilding indexes can be time-consuming and error prone. A Microsoft Engineer has provided a stored procedure that builds a list of indexes in the database and rebuilds or reorganizes them as appropriate.

The code can be found here:

https://raw.githubusercontent.com/yochananrachamim/AzureSQL/master/AzureSQLMaintenance.txt

After you create the stored procedure, run as follows:

```
EXEC AzureSQLMaintenance 'index'
```

***Note:*** *We recommend you inspect and change the code as required. Run the procedure when the database is not busy. The procedure is processor and I/O intensive.*

## Schedule Index Rebuilding

Schedule index rebuilding regularly; daily or weekly depending on how quickly indexes are becoming fragmented.

Microsoft supplies Azure Automation to allow the scheduling of stored procedure calls.

For more information, please see Managing Azure SQL databases using Azure Automation.

## Database Backups

Backing up the Privilege Management database on a regular basis is important for preserving Privilege Management activity in the event of a hardware or system failure on the SQL Server that may cause a corruption. Backed up databases can be quickly restored with minimum disruption to the business.

There are several options for backing up components of a database. We recommend backing up the entire database.

Therefore, we recommend a Maintenance Plan is applied to SQL Server to ensure the Privilege Management database is fully backed up on a regular basis.

## Create a Maintenance Plan

Maintenance Plans allow you to create a workflow of maintenance tasks in SQL Server to ensure your databases are fully optimized and backed up. Plans can be created manually, or by using the built-in wizard, and can be performed manually or automatically on a schedule.

> ***Note:*** *Maintenance Plans are executed as SQL Server Agent jobs. The SQL Server Agent must be running.*

To create a Maintenance Plan in SQL Server Management Studio:

1. Navigate to **Management > Maintenance Plans**.
2. Right-click **Maintenance Plans** and choose **Maintenance Plan Wizard**.
    - Rebuild Index
    - Backup Database (Full)
3. Proceed through the wizard to the **Select Maintenance Tasks** page and check the following recommended tasks (as a minimum):
4. Proceed through the wizard (setting any options as appropriate) to the **Define Rebuild Index Task** page.
5. Select the BeyondTrustReporting Database and click **OK**.
6. Proceed through the wizard (setting any options as appropriate) to the **Define Back Up Database (Full) Task** page.
7. Select the BeyondTrustReporting Database and click **OK**.
8. Set the backup schedule, backup location, and any other options as appropriate.
9. Proceed through the wizard (setting any options as appropriate). Click **Finish** to complete the wizard and create the new Maintenance Plan.

The new plan will now be listed under the **Maintenance Plans** node and can be edited at any time. The Maintenance Plan can be run manually by right-clicking and choosing **Execute**.

## Purge Privilege Management Data

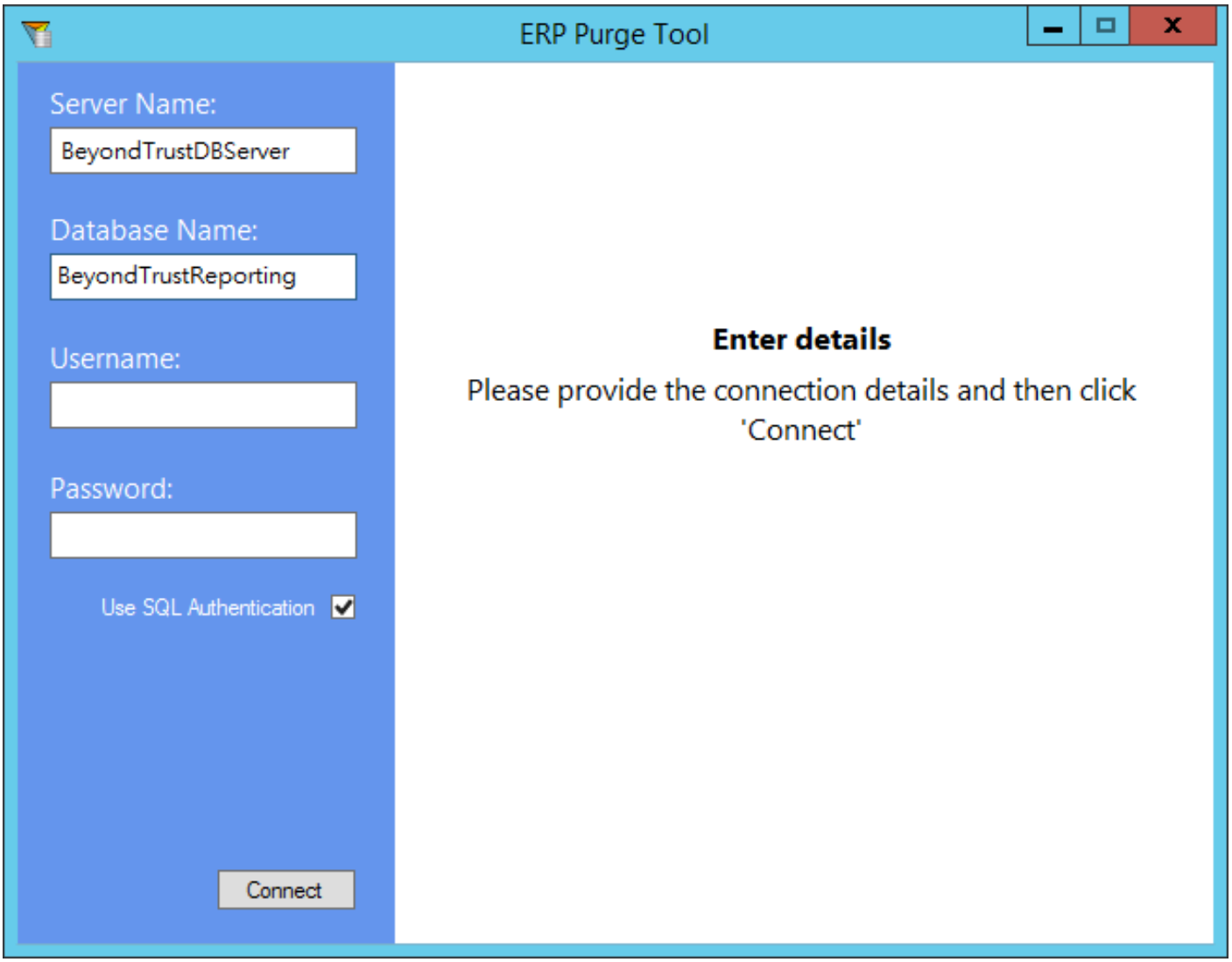
Privilege Management Reporting includes an optional **ER Purge Tool**, which allows old data to be purged from the Privilege Management database. The ER Purge Tool can be downloaded from the BeyondTrust website. After you install the ER Purge Tool, it can be run from the Windows Start Menu.

> ***Note:*** *Before purging large sets of data, please ensure your SQL Transaction logs can grow to accommodate. It may be necessary to delete data in stages when setting this up for the first time.*

### Connect to Privilege Management Reporting

When launched, you must connect to the database you want to purge. Enter the name or IP Address of the SQL server instance and the name of the Privilege Management Database.

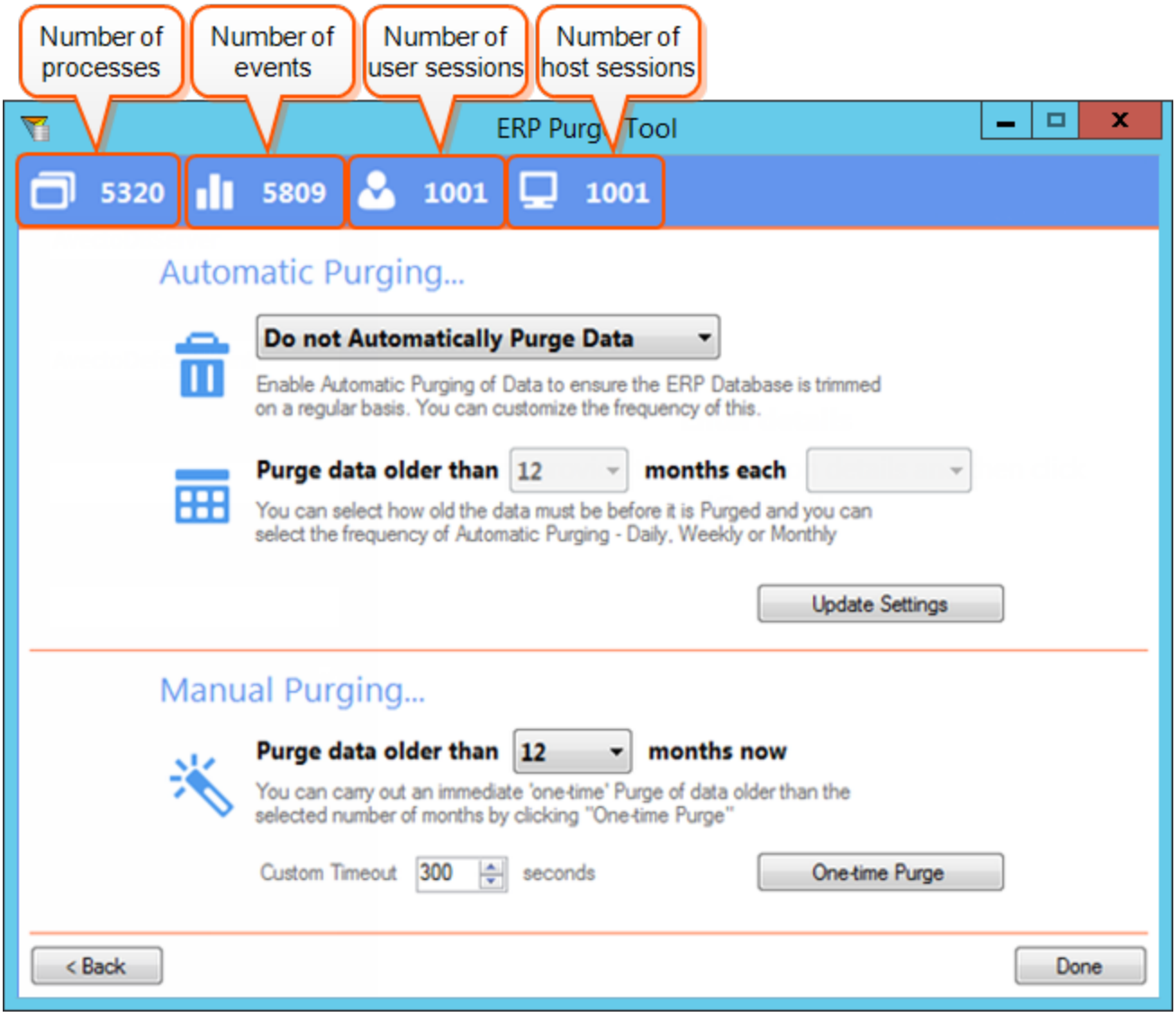
The tool supports either pass through Windows authentication, or you can select a SQL account and password by checking the **Use SQL Authentication** box.




> ***Note:*** *To use the ER Purge Tool, connect to SQL Server with* ***db_owner*** *privileges for the* ***Privilege Management*** *database and the* ***msdb*** *database.*

For convenience, successful connection settings (not including the password) are recalled the next time the tool runs.

Once connected, the tool starts to collect summary statistics for the database. These statistics are collected in the background, and depending on the size of the database, may take several seconds to appear.




***Note:*** *The total number of events and processes may not be the same. This is expected as some actions, such as Privilege Monitoring may generate multiple events for the same process.*

When a purge is performed, then any data relating to processes, events, user sessions, and host sessions older than the specific period of months will be deleted.

The ER Purge Tool offers two modes of configuration:

- **Automatic Purging**
- **Manual Purging**

## Automatic Purge

> ***Note:*** *Before purging large sets of data, please ensure your SQL Transaction logs can accommodate. It may be necessary to delete data in stages when setting this up for the first time.*

The ER Purge Tool allows you to configure an automatic purge job using the SQL Server Agent. If there is an existing job configured on the SQL Server instance, then the current job settings will be automatically populated, otherwise automatic purging will default to **Do not automatically purge data**.

To configure an automatic purge:

1. Expand the drop-down box and choose **Automatically Purge Data**.
2. Select the number of months (1 to 12) for which older data should be purged.
3. Set the frequency for automatic purges (each day, each week or each month).
4. Click **Update Settings** to create or update the automatic purge.

> ***Note:*** *To configure an automatic purge, the SQL Server Agent must be running.*

Once configured, the SQL Server Agent will automatically purge data according to the age and frequency set.

- **Daily**: The purge occurs at 00:00:00 each day.
- **Weekly**: The purge occurs at 00:00:00 each Monday.
- **Monthly**: The purge occurs at 00:00:00 each first day of the month.

For advanced SQL administrators, you can edit the SQL Server Agent job in SQL Server Management Studio to change the purge schedule. However, you may receive the following error:

```
Creating an instance of the COM component with CLSID {AA40D1D6-CAEF-4A56-B9BB-D0D3DC976BA2} from the
IClassFactory failed due to the following error: c001f011. (Microsoft.SqlServer.ManagedDTS)"
```

This is a known Microsoft issue.

> For more information, please see >https://support.microsoft.com/kb/2315727 for steps to resolve this issue.

## Manual Purge

> ***Note:*** *Before purging large sets of data, please ensure your SQL Transaction logs can grow to accommodate. It may be necessary to delete data in stages when setting this up for the first time.*

If at any time you want to manually purge Privilege Management data, you can use **Manual Purging**. To configure a manual purge:

1. Set the number of months (1 to 12) for which older data should be purged.
2. Select a timeout in seconds for how long the purge task runs before it is terminated. The default timeout is **300** seconds.
3. Click **One-time Purge** to activate the manual purge.
4. At the confirmation prompt, click **Yes** to confirm.

A confirmation is displayed after the purge finishes.

***Note:** The ER Purge Tool must remain running for the duration of the manual purge.*

## Purge Data by Stored Procedure

You can use the Stored Procedure **PurgeData** to purge data from Privilege Management. PurgeData accepts two arguments, you use one or the other but not both:

- **Date**: All events from before this date are purged.
- **Integer**: All events older than this number (in months) are purged.

### Examples

Below are two examples using the PurgeData stored procedure.

```
EXEC PurgeData ('20170501', NULL)
```

This purges all data before the specified date in the format **YYYYMMDD**.

```
EXEC PurgeData (NULL, 6)
```

This purges all data older than 6 months.

### Small Batch Examples

It may be necessary to delete data in smaller batches than above so the transaction log does not fill up. Only use this approach if there is a backlog of data to purge.

```
EXEC PurgeData (NULL, 12);
EXEC PurgeData (NULL, 11);
EXEC PurgeData (NULL, 10);
EXEC PurgeData (NULL, 9);
EXEC PurgeData (NULL, 8);
EXEC PurgeData (NULL, 7);
EXEC PurgeData (NULL, 6);
```

This purges all data older than 12 months, then 11 months, 10 months, and so on.

Use the first parameter if you need to delete in smaller batches than 1 month:

```
EXEC PurgeData('20180101', NULL)
EXEC PurgeData('20180102', NULL)
EXEC PurgeData('20180103', NULL) etc
```

This purges all data before the specified date in the format **YYYYMMDD**.

## Purge by Individual User, Host, or Workstyle

You can use the stored procedure **PurgeEventsByEntity** to purge events from a specified Host, User, or Workstyle by running the script below with the specified ID.

You can query the **Hosts**, **FullDetail_Users**, or **Policies** tables for the **HostID**, **UserID**, or **Workstyle** name respectively. The ID is populated in the "Script" on page 40 to purge the events from that specific entity. Only one ID can be used each time you run the script.

To obtain the HostID from the Hosts table:

```
SELECT HostID FROM Hosts WHERE NAME = 'YourHostName'
```

To obtain the HostID from the Users table:

```
SELECT UserID FROM FullDetail_Users WHERE FormattedName = 'YourDomain\YourUser'
```

To obtain the PolicyID:

```
SELECT ID FROM Policies WHERE PGPolicyName = 'WorkstyleName'
```

### Script

Only one ID can be used each time you run this script. Replace the other two IDs with 'NULL'.

```
EXEC PurgeEventsByEntity HostID, UserID, PolicyID
```

### Examples

To purge events for HostID 12, run:

```
EXEC PurgeEventsByEntity 12, NULL, NULL
```

To purge events for UserID 17, run:

```
EXEC PurgeEventsByEntity NULL, 17, NULL
```

To purge events for Workstyle 5, run:

```
EXEC PurgeEventsByEntity NULL, NULL, 5
```

## Shrink the Database

If a large amount of data is being purged from the Privilege Management database, we recommend the database is shrunk once the purge is complete. Shrinking the Database reduces the disk space consumed by the database and log files by removing empty data and log pages.

A database shrink can be configured as a Maintenance Plan in SQL Server Management Studio, and can be configured to run on a regular schedule.

# Database Population

Event data enters the database using four staging tables:

- **Staging** for process type events
- **Staging_UserLogon** for user logon events
- **Staging_ServiceStart** for service start events
- **Staging_ServiceStop** for service stop events

The data in the tables is normalized across the database using the **CopyFromStaging** stored procedure.

This procedure is normally called every 10 seconds by a Service Broker contract. This is setup by the database installer.

## CopyFromStaging Locks

The **CopyFromStaging** stored procedure requires exclusive access to the staging data otherwise data could be lost. To prevent it being run concurrently in separate sessions a configuration table called Config manages which process currently has a lock on the system.

A row in the table has a configid of **CopyFromStagingLocked** and a BitValue. If the BitValue is set to 1 the **CopyFromStaging** stored procedure terminates without processing any events. The StringValue column is used to show what has a lock on the system.

## Restart the Database

If the database is restarted when **CopyFromStaging** is running the lock table remains in place.

Please use the following procedure before restarting the database (or rebooting the database host).

1. Prevent **CopyFromStaging** from being called by renaming it.
2. If **CopyFromStaging** is currently executing you should wait for it to complete. If it is still executing then the Config table will show that the BitValue column equals 1 where the ConfigId is **CopyFromStaging**. If you instructed **CopyFromStaging** to run for a certain amount of time by setting the **@MinDurationToRunForInMinutes** parameter you can stop **CopyFromStaging** after the next batch by executing **InterruptCopyFromStaging**.

> ***Note: CopyFromStaging****processes a batch of 10000 events at a time. It checks the new events against existing events for duplicates so if you have a large database, then the processing of each batch can take some time.*

3. Restart the database / machine.
4. Rename **CopyFromStaging** back to its original name.

## Recover from a Restart Leaving the Lock in Place

If you restarted the database without following the above procedure, then the lock may remain in place and **CopyFromStaging** will terminate without processing any events. If you are sure that no events are being processed, then you can delete the lock by executing the procedure **ReleaseStagingLockCopyFromStaging**. This will result in any data in the current batch being lost. If you want to recover without losing the current data, please contact BeyondTrust Technical Support.

## Database Error Management

If an error occurs during the execution of **CopyFromStaging** the batches in **StagingTemp** and the three other **StagingTemp_x** tables are copied to **StagingTempBadBatches** and **StagingTemp_xBadBatches** and the error message is stored in the **StagingErrors** table. Processing of new events will then continue as normal.

If a batch has an error then the whole batch is copied to **StagingTempBadBatches**, not just the bad rows. To process the data in **StagingTempBadBatches** and leave just the bad rows you can call **RetryCopyFromStaging**. This will process the rows one at a time and leave only the offending ones.

If **CopyFromStaging** is running, then **RetryCopyFromStaging** will not run. If you are using the Service Broker you could temporarily rename **CopyFromStaging** to stop it being run again. If you are using jobs you can disable the job.

If **CopyFromStaging** is set to run for a long time you can call **InterruptCopyFromStaging** to stop **CopyFromStaging** after the current batch is processed.

# Event Management and Behavior

## Event Parser SQL Connection

The connection between the Event Parser and the Database is established using Microsoft SQL Native Client (OLE DB) provider.

For more information, please see SQL Server Native Client (OLE DB).

- The connection is secured using Windows Authentication.
- The Event Parser runs as a Windows service using user credentials that have access to insert data to the Privilege Management Reporting database.
- The connection is established when the first event is processed, and remains open thereafter. If the connection breaks while executing commands, the parser tries to recreate the connection. Data will not be lost due to an occasional loss of connection.

## Data Transmission

The Event Parser service processes audit events in the shortest time possible, using a batching approach.

***Note:*** *The number of events processed in each batch is not configurable in the current release.*

The Event Parser “subscribes” to the Event Log and is notified of new events.

When the Event Parser is notified new data is available, all events available are processed in batches of 100.

Audit data are inserted to the Privilege Management Reporting database using bulk SQL insert to optimize performance.

The Privilege Management Reporting SQL database is designed to eliminate duplicate audit data, so there is no need to roll back partial failures; transactional inserts are not used.

If the data insert fails, the Event Parser continues to retry; it will not skip over events.

For example, if the Event Parser Service’s account password expires, the Event Parser will fail to establish or reconnect to the database and “get stuck” retrying the same insert until the condition is rectified. This is by design, to ensure no data is lost.

If the failure persists for an extended period, the Windows Event Log may begin to “roll over” causing the oldest audit events to be removed. Be sure to maximize the event log size, and monitor growth rate to ensure audit data is retained as long as necessary.

## Monitor and Recovery

To diagnose failures in the Event Parser service look in the Windows Application event log on the Windows Event Collector host.

The Event Parser service will raise events if errors occur, such as failure to connect to the database. These events typically contain information required to diagnose the problem. If this is insufficient, debug logging can be enabled. The debugging logs are designed for advanced diagnostics by BeyondTrust staff.

Please open a support case.

## Reprocess Data

If data needs to be reprocessed (for example, the database is deleted and recreated), the Event Parser can reparse the entire event log. This is always safe to do, as the database is fully resilient to duplicate data being added; duplicate data is discarded.

Be aware that reprocessing all the events creates a lot of database activity in a short period of time. It is best to plan this during periods of low activity in your environment.

To do this:

1. Stop the BeyondTrust Privilege Management Event Parser service.
2. Delete the registry key:

```
HKEY_USERS\<Event Parser User SID>\Software\Avecto\Privilege Guard Event Parser\
```

3. Start the BeyondTrust Privilege Management Event Parser service.