



ADMINISTRATOR GUIDE

# Database Performance Analyzer

Version 2020.2

© 2020 SolarWinds Worldwide, LLC. All rights reserved.

This document may not be reproduced by any means nor modified, decompiled, disassembled, published or distributed, in whole or in part, or translated to any electronic medium or other means without the prior written consent of SolarWinds. All right, title, and interest in and to the software, services, and documentation are and shall remain the exclusive property of SolarWinds, its affiliates, and/or its respective licensors.

SOLARWINDS DISCLAIMS ALL WARRANTIES, CONDITIONS, OR OTHER TERMS, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, ON THE DOCUMENTATION, INCLUDING WITHOUT LIMITATION NONINFRINGEMENT, ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION CONTAINED HEREIN. IN NO EVENT SHALL SOLARWINDS, ITS SUPPLIERS, NOR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, WHETHER ARISING IN TORT, CONTRACT OR ANY OTHER LEGAL THEORY, EVEN IF SOLARWINDS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The SolarWinds, SolarWinds & Design, Orion, and THWACK trademarks are the exclusive property of SolarWinds Worldwide, LLC or its affiliates, are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other SolarWinds trademarks, service marks, and logos may be common law marks or are registered or pending registration. All other trademarks mentioned herein are used for identification purposes only and are trademarks of (and may be registered trademarks) of their respective companies.

# Table of Contents

<b>DPA Introduction</b> .....	<b>7</b>
Introduction to DPA .....	7
DPA architecture .....	9
<b>DPA licensing</b> .....	<b>11</b>
DPA license types .....	11
DPA registration and licensing options for clustered environments .....	13
Requirements for monitoring a database instance running in a VM cluster .....	15
Purchase and view DPA licenses .....	16
Activate DPA licenses .....	17
Allocate or deallocate DPA licenses .....	19
Troubleshoot over-allocated DPA licenses .....	21
Deactivate your DPA licenses .....	22
<b>Register a database instance for monitoring with DPA</b> .....	<b>23</b>
Database instances DPA can monitor .....	24
Register multiple database instances .....	30
Register an Oracle database .....	32
Register a SQL Server database .....	37
Register a Sybase database .....	41
Register a Db2 database instance .....	43
Register a MySQL, Percona, or Maria database .....	46
Register a PostgreSQL database instance and prepare for monitoring .....	50
Register an Amazon RDS for Oracle database .....	56
Register an Amazon RDS for SQL Server database .....	59
Register an Amazon RDS for MySQL or Aurora database instance .....	61
Register an Azure SQL database .....	66

Register an Azure SQL Managed Instance .....	69
Unregister a monitored database instance .....	71
<b>Database instance groups .....</b>	<b>73</b>
About monitoring SQL Server Availability Groups with DPA .....	73
About monitoring Oracle multitenant databases (CDBs) .....	78
Manually group database instances in DPA .....	79
View information about a group of database instances .....	81
<b>Monitor database instances with DPA .....</b>	<b>82</b>
Update a monitored database instance .....	83
Stop monitoring a database instance for a period of time .....	83
DPA troubleshooting tips .....	84
<b>Investigate performance issues with DPA .....</b>	<b>86</b>
Access DPA query or table tuning advisors .....	86
Use DPA's query performance analysis to find the root cause of performance issues .....	87
Investigate inefficient queries running against a table .....	93
Table tuning best practices .....	102
Identify blocking sessions and deadlocks with DPA .....	104
Find and investigate unusually long wait times (anomalies) .....	109
About anomaly detection in DPA .....	118
Add an annotation to document a change to the database .....	121
<b>Manage SQL statements .....</b>	<b>124</b>
Name SQL statements .....	124
Exclude SQL statements from DPA .....	126
Add excluded SQL statements back to DPA trend charts and analysis .....	133
<b>Resource metrics in DPA .....</b>	<b>135</b>
View resource metrics in DPA .....	135
Show or hide VMware events on metric charts .....	137

About DPA resource metric baselines .....	138
View or change DPA resource metric thresholds .....	140
Metrics collected by DPA .....	141
<b>DPA user accounts .....</b>	<b>189</b>
DPA roles and privileges .....	189
Create a DPA user account and assign privileges .....	191
DPA user authentication options .....	192
Configure DPA to use Active Directory or LDAP .....	193
<b>DPA alerts .....</b>	<b>197</b>
View the status and history of DPA alerts .....	197
Configure a DPA Wait Time alert .....	200
Configure a DPA Resources alert .....	203
Configure a DPA Administrative alert .....	206
Configure a DPA Custom alert .....	209
Send SNMP traps from DPA alerts .....	221
Stop DPA alerts for a period of time .....	222
Create a DPA alert group .....	224
Create and manage DPA contacts and contact groups .....	225
Notification policy for DPA alerts .....	227
<b>Define email templates for alert notifications .....</b>	<b>229</b>
Create or edit a custom email template for DPA alert notifications .....	229
Delete a custom email template .....	235
Change the default email template for DPA alert notifications .....	236
<b>DPA reports .....</b>	<b>238</b>
About DPA reports .....	238
Access and run DPA reports .....	239
Create a DPA report .....	240

Search for a SQL statement to report on .....	243
Schedule a DPA report for email delivery .....	246
Create a DPA report group .....	248
<b>Link together separate DPA servers .....</b>	<b>249</b>
Set up a Central Server and add remote DPA servers .....	249
Configure authentication for the DPA Central Server .....	250
View information from remote servers on the DPA Central Server .....	251
Advanced configuration for the DPA Central Server .....	251
<b>Use the DPA REST API .....</b>	<b>256</b>
Manage tokens used for authentication to the DPA API .....	256
Learn about and experiment with the DPA API .....	258
Examples of using Python scripts to make DPA API calls .....	265
Examples of PowerShell scripts that make DPA API calls .....	302
<b>View and manage trusted certificates .....</b>	<b>339</b>
Manage trusted certificates in the DPA trust store .....	339
View trusted certificates in the Java trust store .....	343
Manage DB certificates .....	344
<b>DPA administrative tasks .....</b>	<b>349</b>
Stop and start DPA .....	349
Set advanced DPA configuration options .....	350
Enable SNMP Monitoring in SCOM .....	350
Configure password protection for DPA features that allow custom SQL .....	351

# DPA Introduction


See the following topics to get an overview of DPA features and learn about DPA architecture:

- [Introduction to DPA](#)
- [DPA architecture](#)

## Introduction to DPA

You can use Database Performance Analyzer (DPA) to monitor, diagnose, and resolve performance problems for [many types](#) of database instances, both self-managed and in the cloud.

DPA has [agentless architecture](#) and uses [wait-based analytics](#) for extended database monitoring. DPA uses less than one percent of resources on production systems.

 Get a walk-through of DPA functionality from the [DPA Getting Started Guide](#).

See the following topics to start using DPA:

---

Get the **licenses** you need, and then **register** the databases you want to monitor.

- [DPA license types](#)
  - [Purchase and view DPA licenses](#)
  - [Activate DPA licenses](#)
  - [Allocate or deallocate DPA licenses](#)
  - [Register a database instance for monitoring with DPA](#)
-

Investigate **performance issues** with DPA.

- [Access DPA query or table tuning advisors](#)
- [Use DPA's query performance analysis to find the root cause of performance issues](#)
- [Investigate inefficient queries running against a table](#)
- [Table tuning best practices](#)
- [Identify blocking sessions and deadlocks with DPA](#)
- [Find and investigate unusually long wait times \(anomalies\)](#)
- [About anomaly detection in DPA](#)
- [Add an annotation to document a change to the database](#)

Use **alerts** to become aware of issues and address them proactively before they affect end users.

- [Configure a DPA Wait Time alert](#)
- [Send SNMP traps from DPA alerts](#)
- [Create a DPA alert group](#)
- [Create or edit a custom email template for DPA alert notifications](#)

Use **reports** to identify database trends, track the results of your performance tuning, and communicate those results to others.

- [Create a DPA report](#)
- [Schedule a DPA report for email delivery](#)
- [Create a DPA report group](#)

Manage DPA **user accounts**.

- [DPA roles and privileges](#)
- [Create a DPA user account and assign privileges](#)

For large or geographically separate environments, **link DPA servers** together.

- [Set up a Central Server and add remote DPA servers](#)
- [Configure authentication for the DPA Central Server](#)
- [View information from remote servers on the DPA Central Server](#)



Automate tasks with the DPA REST API.

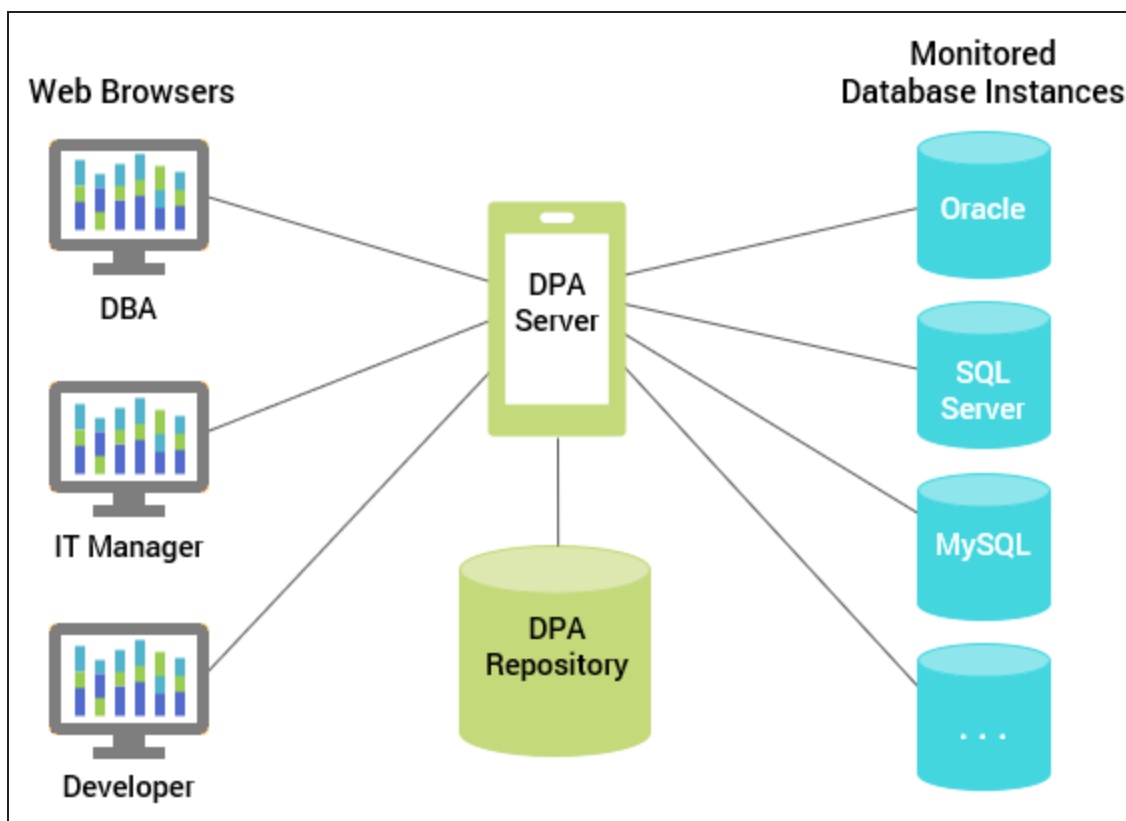
- [Manage tokens used for authentication to the DPA API](#)
- [Learn about and experiment with the DPA API](#)
- [Examples of using Python scripts to make DPA API calls](#)
- [Examples of PowerShell scripts that make DPA API calls](#)

## DPA architecture

Database Performance Analyzer consists of:

- A DPA server
- A DPA repository database
- One or more database instances you want to monitor

The DPA server collects performance data from a set of database instances you choose to monitor. DPA stores this data in the repository database, and makes it available to users through its web-based interface.



For optimal performance, the DPA server, repository database, and the monitored database instances must reside on the same high-speed LAN. If your environment contains database instances that are on separate LANs, SolarWinds recommends [creating a repository database](#) on each LAN. For cloud monitoring, SolarWinds recommends deployment to the same region.

SolarWinds recommends installing one DPA instance on a computer. If you must install multiple instances on the same computer, [submit a support ticket](#).

## Key functions of the DPA server

The DPA server:

- Collects data from the monitored database instances and stores the data in the repository database.
- Provides a web-based interface that displays performance data from any computer with access to the DPA server. From this interface, you can monitor database activity, investigate performance issues, and configure alerts and reports.

## Agentless monitoring for database instances and virtual environments

DPA remotely connects to each database instance using Java Database Connectivity (JDBC). DPA causes less than 1% overhead on the instance. No software is installed on the monitored server.

In a virtual environment, DPA can remotely connect to each VMware vCenter Server, ESX, or ESXi host. DPA causes less than 1% overhead on the monitored systems. No software is installed in the vCenter Server, ESX or ESXi host, or virtual machines.

DPA runs a query (the "quickpoll" query) on monitored database instances to collect information about wait events. By default, the quickpoll query runs once per second.

# DPA licensing

See the following topics to learn more about DPA licensing:

- Learn about [DPA license types](#) and subscription licensing for DPA servers deployed in the Amazon Web Services (AWS) Marketplace.
- Learn about [registration and licensing options for clustered environments](#).
- If you are monitoring a database instance that runs in a VM cluster, see the [requirements to create a user](#).
- [Purchase licenses](#) and view your purchased licenses in the Customer Portal.
- [Activate DPA licenses](#) to make them available to a DPA server.
- [Allocate licenses](#) to the DPA database instances you want to monitor, or deallocate a license to make it available to another instance.
- If licenses are over-allocated, [troubleshoot](#) and resolve the issue.
- [Deactivate DPA licenses](#) to make them available to a different DPA server.

## DPA license types

DPA provides the following licensing options:

- **Subscription licensing** is used for DPA servers deployed in the Amazon Web Services (AWS) Marketplace.
- For other DPA servers, an **individual license** is required for each monitored database instance.

## Individual licenses

SolarWinds sells individual licenses by category according to the database edition they are authorized to monitor. You must [purchase an individual license](#) for each database instance you monitor. In addition, you can purchase virtual machine licenses to monitor the virtual infrastructure hosting a database instance.

License Type	Can Monitor
Category 1 licenses	Category 1 licenses can monitor all database types. They are <b>required</b> for: <ul style="list-style-type: none"> <li>• Oracle: all editions except Standard and Express</li> <li>• Sybase: all editions except Express</li> <li>• Db2: all editions except Express</li> </ul>

License Type	Can Monitor
Category 2 licenses	<ul style="list-style-type: none"> <li>• Oracle: Standard and Express editions (including Amazon RDS)</li> <li>• SQL Server: all editions (including Amazon RDS)</li> <li>• MySQL, Percona, or MariaDB: all editions (including Amazon RDS and Aurora)</li> <li>• PostgreSQL or EDB Postgres: all editions (including Amazon RDS and Aurora)</li> <li>• Sybase: Express edition</li> <li>• Db2: Express edition</li> </ul> <p>If you run out of Category 2 licenses, you can use Category 1 licenses instead.</p>
Azure SQL Database license	<ul style="list-style-type: none"> <li>• Azure SQL Database: all editions, including databases in elastic pools</li> </ul> <p>If you run out of Azure SQL Database licenses, you can use Category 1 or Category 2 licenses instead.</p>
VM Option licenses	<p>If a database instance runs on a virtual machine (VM), you can apply an optional VM license in addition to the Category 1 or Category 2 license. When you apply a VM license, DPA collects performance metrics from the VM and the physical host on which the database instance runs. This information is displayed in the Virtualization view.</p>

## All individual licenses are floating

You can register more instances than you have licenses for. On the license allocation page, assign the licenses to the instances you want to monitor.

DPA does not collect data from registered database instances that are not licensed. However, you can view previously collected data on those database instances.

## Clustered environments

For information about registering SQL Server AGs and Oracle RACs, see [Registration and licensing options for clustered environments](#).

If you are monitoring a database instance that runs in a virtual machine (VM) cluster, a [user with at least read-only permissions is required](#) on the hosts and VMs that will be monitored.

## Subscription licensing

When you deploy the DPA server from the AWS Marketplace, DPA uses subscription licensing. As you register databases and monitor them, DPA charges your Amazon Subscription through the AWS Metering Service. DPA charges are based on the number of database instances you monitor each hour. See the AWS Marketplace for details and pricing.

With subscription licensing, you can monitor any supported database type (like the Category 1 license described in the following section). However, you cannot access the VM-related information that is available with a VM Option license.

**i** If you want to use individual DPA licenses in the Amazon cloud, you can deploy an EC2 instance, install DPA, and apply your licenses. You cannot use both individual DPA licenses and a subscription on a single DPA server.

## Learn more

For more information about purchasing and allocating licenses, see:

- [Purchase and view licenses](#)
- [Activate individual DPA licenses](#)
- [Allocate or deallocate individual DPA licenses](#)
- [Deactivate your licenses](#)
- [Troubleshoot over-allocated licenses](#)

## DPA registration and licensing options for clustered environments

To get the maximum value from DPA, SolarWinds recommends the following options for registering and licensing SQL Server Availability Groups (AGs) and Oracle Real Application Clusters (RACs).

**i** Every environment is different, so talk with your SolarWinds representative for other options.

### SQL Server AGs

You can register SQL Server availability groups (AGs) using either of the following options:

- Register each SQL Server instance in the cluster
- Register the AG listener

#### Register each SQL Server instance in the cluster


If there are multiple AGs in the cluster, this option is recommended because it ensures that DPA does not monitor the same instance more than once. DPA monitors all activity on each instance, including primary and secondary AG activity.

With this option, DPA does not follow AGs when they fail over. Monitoring all instances in the cluster ensures that you see all activity when AG failovers occur.

## Register the AG listener

Use this option if you want to monitor activity on the instance that contains the primary replica of an AG. When the AG fails over, DPA follows the listener and begins monitoring the SQL Server instance that now acts as the AG's primary replica.

SolarWinds recommends registering only one listener per cluster unless you can ensure that no instance in the cluster will act as the primary replica for multiple AGs. If you register multiple listeners and the same instance acts as the primary replica for more than one of the AGs, DPA monitors that instance multiple times. Duplicate monitoring is not recommended.

 SQL Server logins are **not** automatically replicated. To enable DPA to continue monitoring after a failover, you must [manually create the DPA login on all instances](#) in the cluster that can act as the primary replica for the AG.


## Oracle RACs

For Oracle RAC (Real Application Clusters), register every instance in the cluster. Do not register the virtual IP that distributes load across the RAC instances.

For Oracle RAC with Data Guard, register both environments but only monitor the primary one. If a failover occurs, simply reassign the licenses to the instances in the secondary RAC environment.

When you register a RAC instance, listener configuration changes might be needed if you are not listening on the physical IP address. SolarWinds recommends:

- If you are registering pluggable databases (PDBs) on a RAC instance, register with the physical IP address of the host.

 To monitor an Oracle multitenant container database (CDB), register each PDB contained in the CDB. You cannot register the CDB directly.

When you register two or more Oracle PDBs in the same CDB, DPA automatically creates a group for the CDB. For more information, see [About monitoring Oracle multitenant databases \(CDBs\)](#).

- If you are registering a non-PDB RAC instance, register with the SID.
- If you are using the Service Name, use the physical IP address of the host. Do not use the virtual IP address (VIP) or the Oracle Single Client Access Name (SCAN) IP address.

## Learn more

For more information about licensing, see the following topics:

- [License types](#)
- [Purchase and view licenses](#)
- [Activate individual DPA licenses](#)
- [Allocate or deallocate individual DPA licenses](#)

## Requirements for monitoring a database instance running in a VM cluster

If you are using DPA to monitor a database instance that runs in a virtual machine (VM) cluster, a user with at least read-only permissions is required on the hosts and VMs that will be monitored. The monitored hosts and VMs include all of the following:

- The VMs that monitored database instances are running on.
- All hosts that those VMs could potentially run on (for example, all hosts in a DRS cluster).
- Other VMs on those hosts.

SolarWinds recommends giving the user read-only permissions on the entire vCenter Server or ESX/ESXi host so that DPA can dynamically monitor any entity as system changes take place.

## Create a user on the vCenter Server or the ESX/ESXi host

Before you can assign user permissions, you must create the user:

- vCenter Server user: Authorized users for vCenter Server are those included in the Windows domain list referenced by vCenter Server or local Windows users on the vCenter Server system. To edit the user list or change user passwords, use the tools you use to manage your Windows domain or Active Directory.
- ESX/ESXi host user: Log in to an ESX/ESXi host as root using the vSphere Client. Then use the Users and Groups tab to add users, remove users, change passwords, set group membership, and assign the required permissions.


## Assign user permissions to inventory objects

Use the vSphere Client to assign user permissions to inventory objects, such as the vCenter server, data center, host, or folder. Requirements and best practices:

- You must have modify permission on an object to be able to assign permissions to that object.
- SolarWinds recommends selecting the entire vCenter Server or ESX/ESXi host and assigning permissions to it.
- Make sure that the Propagate to Child Objects option is selected. This ensures that when new objects are inserted in to the inventory hierarchy, they inherit the required permissions.


## Purchase and view DPA licenses

For DPA servers that use [individual licenses](#), DPA has a 14-day evaluation license. During the evaluation period, you can monitor and view data for an unrestricted number of database instances. After the evaluation period, to continue monitoring you must purchase the correct quantity and [type of licenses](#) for your database instances.

-  • DPA has its own licensing and does not work with SolarWinds License Manager.
- If your DPA server is deployed in the Amazon Web Services (AWS) Marketplace, DPA uses [subscription licensing](#). You do **not** need to purchase, activate, or allocate individual licenses.

## Purchase licenses

Contact our sales team to purchase licenses directly from SolarWinds.

-  Only buy licenses for active database instances. Standby database instances used for disaster recovery or high availability do not need licenses.

- [Online quote tool](#)
- [sales@solarwinds.com](mailto:sales@solarwinds.com)
- 866.530.8100

## View purchased licenses

After you purchase individual licenses, you can view your DPA licenses in the SolarWinds Customer Portal.

1. Access the [Customer Portal](#).
2. Click Licenses > Manage Licenses.




The licenses for your DPA product are listed by [license type](#).

## Next steps

After you have purchased licenses, you must [activate](#) them on a DPA server, [register database instances](#) for monitoring, and then [allocate licenses](#) to the registered instances.

## Activate DPA licenses

After the DPA trial period ends, DPA monitors only licensed instances. If your DPA server uses [individual licenses](#), you must activate a license for each database instance that you want to monitor. Make sure that you have the correct [license types](#) for the database instances you want to monitor.

-  For information about licensing options for SQL Server Availability Groups and Oracle RAC environments, see [Registration and licensing options for clustered environments](#).
- If your DPA server is deployed in the Amazon Web Services (AWS) Marketplace, DPA uses [subscription licensing](#). You do **not** need to purchase, activate, or allocate individual licenses.

## Activate licenses online

If the DPA server is connected to the Internet, you can activate licenses online.

1. Complete the following steps to retrieve your license activation key from the Customer Portal.

If you are evaluating DPA and have received a license activation key from a SolarWinds representative, continue with step 2.

- a. Log in to the [SolarWinds Customer Portal](#).
  - b. Choose Licenses > Manage Licenses.
  - c. Locate the license, and expand it.
  - d. Copy the activation key.
2. On the DPA homepage, click License Management. Then click License Manager.
  3. Click Enter Activation Key.
  4. Select Online Activation, and click Next.
  5. On the Online Activation page, paste the activation key into the correct field.
  6. In the Amount to Activate section, select All Available or Specify Amount.

**i** Unactivated licenses can be activated later. You can reuse an activation key on a different DPA server and activate remaining licenses there.

7. Enter the remaining information, and click Activate.

## Activate licenses offline

Offline activation requires a transfer of files between the DPA server and a computer connected to the Internet. You can use email, shared storage, or a USB flash drive.

1. In DPA, click License Management > License Manager.
2. Click Enter Activation Key.
3. Select Offline Activation, and click Next.
4. Complete the following steps to obtain a license activation file from the Customer Portal.

If you are evaluating DPA and have received a license activation file from a SolarWinds representative, continue with step 5.

- a. On the Offline Activation page, copy the text string next to the license type you want to activate, and save it to a text file. This is your unique machine ID. Include the brackets. For example:

```
[7R12-X2QN-U8XM-WXTD-23H7-0TD7-59QH-6ERF-5BRN-2M17-328G-0DT2-MNMS-005C-000Z-04Q2-0000]
```

- b. Transfer this text file to a computer with Internet access.
  - c. Log in to the [SolarWinds Customer Portal](#).
  - d. Locate the license, and expand it.
  - e. Click Activate license manually.
  - f. Paste the text string into the Unique Machine ID field, and enter the other required information.
  - g. Click Generate License File to download the license file.
  - h. Transfer the license file to the DPA server.
5. On the Offline Activation page, click Choose File and browse to the license file.
  6. Click Activate.

## Next steps

When you activate a license, DPA automatically allocates the license to a registered database instance **if** you have enough licenses to monitor all registered instances in that license category. If you do not have enough licenses to monitor all registered instances, you must [manually allocate licenses](#) to the instances you want to monitor.

## Allocate or deallocate DPA licenses

If your DPA server uses [individual licenses](#), a license must be allocated to each [registered](#) database instance that you want to monitor. DPA starts monitoring new instances immediately after licenses are allocated.

**i** If your DPA server is deployed in the Amazon Web Services (AWS) Marketplace, DPA uses [subscription licensing](#). You do **not** need to purchase, activate, or allocate individual licenses.

[Category 1, Category 2, and Azure SQL Database licenses](#) collect the data shown in the Performance view. VM licenses collect the data shown in the Virtualization view.

## Automatic license allocation

When you [register a database instance](#) or [activate a license](#), DPA determines if it can automatically allocate a license to each database instance. DPA automatically allocates licenses if there are enough activated licenses to cover all of the database instances in that [license category](#). DPA automatically allocates VM licenses if there are enough VM licenses to cover all database instances that:

- Are linked to a VM
- Have been allocated a Category 1 or 2 license

If you have not activated enough licenses to cover all instances that require that license type, DPA does not allocate any of the licenses. You must manually allocate licenses to the database instances you want to monitor.

Example 1:

1. You register **10** Oracle Enterprise Edition database instances, which require Category 1 licenses.
2. You activate **15** Category 1 licenses.

Result: DPA automatically allocates **10** of the licenses to the Oracle Enterprise Edition database instances.

3. You register **5** additional Oracle Enterprise Edition database instances.

Result: DPA automatically allocates the remaining **5** Category 1 licenses.

### Example 2:

1. You register **15** MySQL database instances, which require Category 2 (or greater) licenses.
2. You activate **10** Category 2 licenses. No Category 1 licenses are available.

Result: DPA does not allocate any of the licenses. You can either activate at least 5 additional licenses, or manually allocate licenses to the instances you want to monitor.

**i** Instances without a license allocated to them remain registered with DPA, and you can view performance data that was collected in the past. You can deallocate a license from one registered instance and allocate it to another if necessary.

## Manually allocate licenses to database instances

Use License Allocation to configure how your licenses are allocated to database instances.

### View current license allocation

1. On the DPA homepage, click License Management.
2. See the current license allocations in the summary boxes near the top of the License Allocation page.

### Allocate licenses to database instances

1. On the License Allocation page, find the database instance you want in the list of registered database instances.
2. Select the Cat 1, Cat 2, or Azure check box next to the instance.
3. Click Save.

The license count is updated after you allocate a license.

## Allocate VM licenses to VM database instances

If a database instance runs on a virtual machine (VM), you can allocate a VM license to it in addition to a Category 1 or 2 license. When you allocate a VM license, DPA collects performance metrics from the VMware system (vCenter Server or ESX/ESXi Host) on which the database instance runs.

1. On the License Allocation page, locate a VM-hosted database instance that has a Category 1 or 2 license allocated to it.
2. Select the VM check box next to the instance.
3. Click Save.

**i** If you are monitoring a database instance that runs in a virtual machine (VM) cluster, a [user with at least read-only permissions is required](#) on the hosts and VMs that will be monitored.

## Deallocate licenses

You can deallocate the license from one database instance to make it available to another database instance.

1. On the DPA homepage, click License Management.
2. Clear the Cat 1, Cat 2, or Azure check box to deallocate licenses.

**i** If you clear a Category 1 or 2 license from an instance that also has a VM license, DPA automatically clears the VM license as well.

## Learn more

For more information about licensing, see the following topics:

- [Purchase and view licenses](#)
- [Activate DPA licenses](#)
- [Troubleshoot over-allocated licenses](#)

## Troubleshoot over-allocated DPA licenses

The DPA homepage displays a red banner if DPA is monitoring more registered database instances than you have licenses to monitor. This can happen in two situations:

- A license expires when you have unexpired licenses of the same type on the server.
- You deactivate a license and have other licenses of the same type on the server.

If DPA licenses are over-allocated, you cannot view or analyze your database instances until you deallocate the extra licenses. DPA continues monitoring the databases, so you will not lose data while you bring the allocated licenses to an allowable level.

**i** If necessary, you can [purchase](#) and [activate](#) additional licenses.

To correct an issue of over-allocated licenses, deallocate database instances until you reach the proper number of licenses. If [Category 2 licenses](#) are over-allocated, assign available Category 1 licenses to cover the shortage. If Azure SQL Database licenses are over-allocated, assign Category 1 or 2 licenses to cover the shortage.

1. On the DPA homepage, click License Management.
2. Locate the over-allocated license type on the allocations chart. Over-allocated license types are shown in red.
3. Clear Cat 1, Cat 2, Azure, or VM check boxes until the chart is no longer red.
4. Click Save.

You should now see your database instances in your views.

## Deactivate your DPA licenses

You can deactivate [individual licenses](#) on a DPA server to make the licenses available elsewhere.

If your DPA server has direct access to the internet, you can deactivate licenses online.

 Evaluation licenses and temporary keys cannot be deactivated.

### Deactivate online

1. On the DPA homepage, click License Management.
2. On the License Allocation page, click License Manager.
3. In the Licenses section, locate the License Key you want to deactivate.
4. Click Deactivate.

### Deactivate offline

To deactivate a license offline in DPA 10.0 or earlier, contact [SolarWinds customer support](#).

To deactivate a license offline in DPA 10.1 and later:

1. From the SolarWinds DPA homepage, click License Management > License Manager.
2. Click Deactivate next to the license.
3. Click Yes to confirm the offline deactivation, and download the deactivation receipt file.
4. Log in to the [SolarWinds Customer Portal](#), and go to the License Management page.
5. Select the DPA instance, and click Deactivate license manually.
6. On the Manage License Deactivation page, browse for the deactivation receipt file, and click Upload.

# Register a database instance for monitoring with DPA

For more information about the database instances you can monitor and the requirements for the privileged user, see [Database instances DPA can monitor](#).

If you are monitoring a **large number** of database instances, use the DPA mass registration feature to [quickly register multiple databases](#).

 You can also register database instances using scripts that call the [DPA API](#).

To register a **single** database instance for monitoring using a wizard, select the type you want to register:

- Self-Managed:
  - [Oracle](#)
  - [Microsoft SQL Server](#)
  - [SAP Sybase ASE](#)
  - [Db2](#)
  - [MySQL, Percona, or Maria](#)
  - [PostgreSQL or EDB Postgres](#)
- Amazon RDS:
  - [Amazon RDS for Oracle](#)
  - [Amazon RDS for SQL Server](#)
  - [Amazon RDS for MySQL or Aurora](#)
  - [Amazon RDS for PostgreSQL](#)
- Azure:
  - [Azure SQL DB](#)
  - [Azure SQL Managed Instance](#)
  - [Azure Database for PostgreSQL](#)

## Database instances DPA can monitor

DPA can monitor database instances you manage on both physical and virtual servers or Amazon RDS instances hosted in the Amazon Elastic Compute Cloud (EC2). DPA can also monitor Azure SQL and Azure SQL Managed Instances. The server hosting DPA must be able to connect to the monitored instance.

### Self-managed databases

**i** For information about the privileges required for the privileged user, see the instructions for [registering each database type](#).

Database	Supported Versions
Oracle	<ul style="list-style-type: none"> <li>• 19 (single tenant and multitenant<sup>1</sup>)</li> <li>• 18.4 (single tenant and multitenant)</li> <li>• 12.2 (single and multitenant)</li> <li>• 12.1 (single and multitenant)</li> <li>• 11.2</li> </ul>
Microsoft SQL Server	<ul style="list-style-type: none"> <li>• 2019 (Windows and Linux)</li> <li>• 2017 (Windows and Linux)</li> <li>• 2016</li> <li>• 2014</li> <li>• 2012</li> </ul> <p>DPA supports the latest SP unless otherwise noted.</p>
SAP Sybase ASE	<ul style="list-style-type: none"> <li>• 16</li> <li>• 15.7</li> <li>• 15.5</li> </ul>
IBM Db2	<ul style="list-style-type: none"> <li>• 11.1</li> <li>• 10.5</li> <li>• 10.1</li> <li>• 9.7</li> </ul>
MySQL <sup>2</sup>	<ul style="list-style-type: none"> <li>• 8.0</li> <li>• 5.7</li> <li>• 5.6</li> </ul>



Database	Supported Versions
Percona <sup>2</sup>	<ul style="list-style-type: none"><li>• 8.0</li><li>• 5.7</li><li>• 5.6</li></ul>
MariaDB <sup>2</sup>	<ul style="list-style-type: none"><li>• 10.4</li><li>• 10.3</li><li>• 10.2</li><li>• 10.1</li><li>• 10.0</li></ul>
PostgreSQL or EDB Postgres	<ul style="list-style-type: none"><li>• 12.x</li><li>• 11.x</li><li>• 10.x</li><li>• 9.6</li></ul>

<sup>1</sup> To monitor an Oracle multitenant container database (CDB), register each pluggable database (PDB) contained in the CDB. Register each PDB just as you would register an Oracle single tenant database. For more information, see [Registration and licensing options for clustered environments](#).

<sup>2</sup>See [Requirements for monitoring MySQL database instances with DPA](#).

## Amazon RDS databases

DPA can monitor the following Amazon RDS instances.

Amazon RDS	Supported Versions
Oracle	<ul style="list-style-type: none"><li>• 19.0</li><li>• 18.0</li><li>• 12.2</li><li>• 12.1</li><li>• 11.2</li></ul>
Microsoft SQL Server	<ul style="list-style-type: none"><li>• 2017</li><li>• 2016</li><li>• 2014 SP1</li><li>• 2012 SP2</li></ul>

Amazon RDS	Supported Versions
MySQL, Percona, or MariaDB	<ul style="list-style-type: none"> <li>• 5.7</li> <li>• 5.6</li> <li>• Aurora 5.7</li> <li>• Aurora 5.6</li> </ul>
PostgreSQL	<ul style="list-style-type: none"> <li>• 11.x</li> <li>• 10.x</li> <li>• 9.6</li> </ul>

### Key differences for Oracle databases on Amazon RDS

Because of Amazon RDS access restrictions, some features that are available on Oracle self-managed database instances are not available for Amazon RDS instances.

Category	Details
Unavailable alerts	Oracle Alert Log Error uses <code>V\$DIAG_ALERT_EXT</code> instead of <code>X\$DBGALERTEXT</code> .
Explain plans	Explain plans cannot be generated with a SYS account. You must specify a different account to generate the live plan.
Workarounds for not having a <code>SYS.UTL_CON</code> package	<ul style="list-style-type: none"> <li>• To kill a real time session, use <code>RDSADMIN.RDSADMIN_UTIL.KILL</code>.</li> <li>• Trace session permissions granted through <code>START_TRACE_IN_SESSION</code> and <code>STOP_TRACE_IN_SESSION</code>.</li> </ul>

### Key differences for SQL Server databases on Amazon RDS

Because of Amazon RDS access restrictions, some features that are available on SQL Server self-managed database instances are not available for Amazon RDS instances.

Category	Details
Unavailable alerts	<ul style="list-style-type: none"> <li>• SQL Server Windows Service Not Running</li> <li>• SQL Server Long Running Jobs</li> <li>• SQL Server Log Has Many Virtual Logs</li> <li>• SQL Server Job Failure</li> <li>• SQL Server Error Log Alert</li> </ul>

Category	Details
Explain plans	The DPA monitoring user does not have a sysadmin role and may have limited access to objects. You can specify a different user to generate the live plan before you generate the plan.
Unavailable metrics	<ul style="list-style-type: none"> <li>• CPU Queue Length</li> <li>• CPU Utilization</li> <li>• Disk Queue Length</li> <li>• Memory Paging Rate</li> <li>• Memory Utilization</li> <li>• Physical I/O Rate</li> <li>• Physical Read Rate</li> <li>• Physical Write Rate</li> </ul>
Workaround for not having a SYSADMIN role	DPA user is a member of PROCESSADMIN role
Deadlock polling	The monitoring user and database administrator (DBA) do not have permission to create a custom Extended Events Session. Only the default <code>system_health</code> Extended Events Session can be used for deadlock polling.

## About repointing database instances

You cannot transfer a registered Oracle or SQL Server database instance between Amazon RDS and a self-managed database and retain DPA historical data. An Oracle or SQL Server database instance transferred between Amazon RDS and a self-managed instance must be registered in DPA as a separate instance.

MySQL database instances can be repointed. After you transfer a MySQL database instance between Amazon RDS and self-managed, you can repoint DPA to the new instance and continue monitoring where you left off. To repoint, use the Update Connection Info wizard in DPA to update the connection details of the registered database instance to point to the new location.

## Azure SQL databases

Database	Required Privileges	Supported Version
Azure SQL	db_owner role	V12
PostgreSQL	N/A	<ul style="list-style-type: none"> <li>• 11.x</li> <li>• 10.x</li> <li>• 9.6</li> </ul>

## Key differences between self-managed SQL Server and Azure SQL database instances

Category	Details
Unavailable Alerts	<ul style="list-style-type: none"> <li>• Transaction Log Freespace</li> <li>• Windows Service Not Running – SQL Server</li> <li>• SQL Server Abnormal Mirroring Status</li> <li>• SQL Server Error Log Alerts</li> <li>• SQL Server Job Failure</li> <li>• SQL Server Log has Many Virtual Logs</li> <li>• SQL Server Long Running Jobs</li> </ul>
Unavailable CPU Metrics	<ul style="list-style-type: none"> <li>• Signal Waits</li> <li>• O/S CPU Utilization</li> </ul>
Unavailable Memory Metrics	<ul style="list-style-type: none"> <li>• Page Life Expectancy</li> <li>• O/S Memory Utilization</li> <li>• Plan Cache Size</li> <li>• Buffer Cache Size</li> <li>• Plan Cache Hit Ratio</li> <li>• Buffer Cache Hit Ratio</li> <li>• Log Bytes Flushed</li> <li>• Log Flushes</li> <li>• SQL Compilation</li> <li>• SQL Re-Compilations</li> </ul>
Unavailable Disk Metrics	<ul style="list-style-type: none"> <li>• Total I/O Wait Time</li> <li>• Total Read I/O Wait Time</li> <li>• Total Write I/O Wait Time</li> <li>• O/S Disk Queue Length</li> <li>• Page Reads</li> <li>• Page Writes</li> <li>• SQL Disk Read Latency</li> <li>• SQL Disk Write Latency</li> </ul>
Unavailable Sessions Metrics	<ul style="list-style-type: none"> <li>• Transaction Rate</li> <li>• Batch Requests</li> </ul>
Unavailable License Compliance Metrics	<ul style="list-style-type: none"> <li>• Core Count</li> </ul>

Category	Details
Additional DTU metrics	<ul style="list-style-type: none"> <li>• DTU Utilization</li> <li>• DTU Consumption</li> <li>• DTU Limit</li> </ul>
Additional Memory metrics	<ul style="list-style-type: none"> <li>• Memory Usage Utilization</li> <li>• XTP Storage Utilization</li> </ul>
Additional Disk metrics	<ul style="list-style-type: none"> <li>• Data I/O Utilization</li> <li>• Log Write Utilization</li> <li>• Database Storage Consumption</li> <li>• Database Size</li> </ul>
Additional Sessions metrics	<ul style="list-style-type: none"> <li>• Max Worker Utilization</li> <li>• Max Session Utilization</li> </ul>

## About repointing database instances

Repointing database instances is not possible between Azure SQL and SQL Server.

## Azure SQL Managed Instance

Database	Required Privileges	Supported Version
Azure SQL Managed Instance (ASMI)	SYSADMIN role	V12

## Key differences between self-managed SQL Server and ASMIs

Category	Details
Unavailable CPU metrics	<ul style="list-style-type: none"> <li>• CPU Queue Length</li> <li>• Instance CPU Utilization</li> </ul>
Unavailable Disk metrics	<ul style="list-style-type: none"> <li>• Physical Read Rate</li> <li>• Physical Write Rate</li> <li>• Physical I/O Rate</li> <li>• O/S Disk Queue Length via WMI</li> </ul>
Unavailable Memory metric	<ul style="list-style-type: none"> <li>• Memory Paging Rate</li> </ul>
Additional Disk metrics	<ul style="list-style-type: none"> <li>• Data I/O Utilization</li> <li>• Log Write Utilization</li> </ul>
Additional Memory metric	<ul style="list-style-type: none"> <li>• XTP Storage Utilization</li> </ul>

Category	Details
Additional Sessions metrics	<ul style="list-style-type: none"><li>• Max Worker Utilization</li><li>• Max Session Utilization</li></ul>

## About repointing database instances


You can repoint a self-managed SQL Server instance to an ASMI. You can use this feature if you are migrating an existing self-managed SQL Server to an ASMI and you want to have DPA data collected from both the self-managed SQL Server and the ASMI associated with the same instance in DPA. However, be aware that ASMIs have different metrics and wait types than SQL Server database instances. Because of these differences, some historical data from the SQL Server database instance will not be displayed after it is repointed to an ASMI.

To retain all data, SolarWinds recommends registering the ASMI as a new instance and reassigning the license from the SQL Server instance. You will still be able to view historical data from the unlicensed SQL Server instance.

 You cannot repoint an ASMI to a self-managed SQL Server instance.

## Register multiple database instances

If you are monitoring a large number of database instances, use the DPA mass registration feature to quickly register multiple databases.

-  • You can also use a wizard to [register a single database instance](#), or you can register database instances using scripts that call the [DPA API](#).
- To register multiple Azure SQL databases using the Mass Registration feature, follow the instructions in [this KB article](#).

Complete the following steps to download a predefined template and enter the required information for all database instances.

1. On the DPA menu, click Options.
2. Under Monitor Setup > Database Instances, click Mass registration.
3. In the Choose a Database Type drop-down, select one of the following:
  - To register different types of database instances, select All database types.
  - To register Azure SQL Managed Instances (ASMIs) or PostgreSQL database instances, select All database types.

- To register database instances that are the same type but **not** ASMIs or PostgreSQL instances (for example, all Oracle or all SQL Server), select the database type.

Choose a Database Type: All database types ▼

- Specify whether you want to edit and save the template on the DPA server or on your local computer.

From local computer
 
 No file chosen

---

From DPA server

Based on your selections, instructions are displayed in the right pane.

- Under How to, click the template link to download the registration file template for the selected database type.

Or, for Azure SQL database instances, click the link to instructions for auto-generating the registration file.

- Edit the file to add information about each database instance:
  - Click the required information link in the How to section for information about what to enter in each column.
  - Do **not** edit the header row.

Type	Notes
Oracle	If you are registering multiple nodes in an Oracle RAC, <a href="#">manually create the monitoring user</a> before you run mass registration. Enter N in the Create Monitoring User column of the mass registration template, and specify the manually created user for all nodes in the RAC.
ASMI	<a href="#">Manually create the DPA monitoring user</a> before you run mass registration, and enter N in the Create Monitoring User column.
PostgreSQL	<a href="#">Manually create the DPA monitoring user</a> before you run mass registration, and enter N in the Create Monitoring User column.  You must also <a href="#">configure the database instances for monitoring</a> before you run mass registration.

- Save the file in `.CSV` format.

**i** If you selected From DPA server, the file **must** be saved in the following location:

```
<DPA_home>/iwc/tomcat/ignite_config/registration
```

Depending on the database type, it must have one of the following file names:

- massreg\_mixed.csv
- massreg\_oracle.csv
- massreg\_sql\_server.csv
- massreg\_azure\_sql\_database.csv
- massreg\_mysql.csv

8. If you selected From local computer, click the Choose File button and select the file you saved.

9. Click Load Registration File.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register an Oracle database

Complete the following tasks to register an individual Oracle database instance for monitoring with DPA.

- i** • If you are monitoring an Oracle Real Application Cluster (RAC), see [Registration and licensing options for clustered environments](#).
- You can use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

## Identify the privileged user

When you register a database instance, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For self-managed Oracle database instances, the privileged user requires the following privileges:

```
SYS user
```



## Complete the registration wizard

1. On the DPA homepage, click Register DB Instance for Monitoring.
2. Under Self-Managed, click Oracle.
3. Click Next.
4. Complete the wizard panels as described in the following table.

Panel	Instructions
Enter Monitored Database Instance Connection Information	<p>DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.</p> <p>Oracle database instances have three connection options:</p> <ul style="list-style-type: none"><li>• Direct Connect</li><li>• Transparent Network Substrate (TNS) Connect Descriptor</li><li>• Lightweight Directory Access Protocol (LDAP) or TNS Name</li></ul> <p>Direct Connect</p> <p>Enter the Service Name or System Identifier (SID), host name or IP address, and port. The default port is 1521.</p> <p>TNS Connect Descriptor</p> <p>The Connect Descriptor value contains everything after <code>NAME=</code> in the <code>tnsnames.ora</code> file. The beginning <code>(DESCRIPTION=</code> is necessary. For example:</p> <pre>(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST = demo.confio.com) (PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = demo)))</pre> <p>LDAP or TNS Name</p> <p>To use this option, Oracle Name Resolution must be configured. For instructions, see <a href="#">Connect to Oracle using name resolution</a>.</p> <p>After you configure Oracle Name Resolution, you can use the LDAP/TNS Name when registering additional monitored database instances.</p> <p>DPA uses the Oracle network configuration <code>.ora</code> files to connect to the database instance.</p> <p>If connection information other than the name has changed, update the <code>.ora</code> files. DPA detects changes to these files automatically. You do not have to update the connection information through this wizard.</p>

Panel	Instructions
Enter Monitored Database Instance Connection Information (continued)	<p data-bbox="381 220 1502 304">If the name has changed, update the <code>.ora</code> files. Then select the check box next to LDAP/TNS Name, and update the value.</p> <p data-bbox="381 325 1502 514">RAC instances</p> <p data-bbox="381 399 1502 514">For an Oracle RAC (Real Application Cluster), register every physical instance in the cluster. Do not register the virtual IP that distributes load across the RAC instances.</p> <div data-bbox="389 546 1510 735" style="border: 1px solid #ccc; padding: 10px;"><p data-bbox="397 556 1502 724"><b>i</b> If you choose to register the virtual IP load balancing listener, or to monitor only a subset of instances in the cluster, DPA will not have complete and consistent data. This will affect DPA's tuning and resource analysis.</p></div> <p data-bbox="381 766 1502 840">For more information, see <a href="#">DPA registration and licensing options for clustered environments</a>.</p> <p data-bbox="381 871 503 913">DBA user</p> <p data-bbox="381 934 1291 976">Enter DBA credentials for DPA to register the database instance.</p>

Panel	Instructions
Enter the Monitoring User	<p data-bbox="383 222 1523 342">Create or specify the account that DPA will use to gather information. To ensure that the account has the required permissions, SolarWinds recommends creating a new account.</p> <p data-bbox="383 373 737 405">To create a new account:</p> <ol data-bbox="415 436 1468 684" style="list-style-type: none"><li data-bbox="415 436 1045 468">1. Next to Create Monitoring User, click Yes.</li><li data-bbox="415 499 959 531">2. Enter the user name and password.</li><li data-bbox="415 562 1468 684">3. Select a Tablespace and Temp Tablespace on the monitored database. This is primarily used for gathering Explain Plan data for monitored queries.</li></ol> <p data-bbox="383 716 824 747">To specify an existing account:</p> <ol data-bbox="415 779 1032 877" style="list-style-type: none"><li data-bbox="415 779 1032 810">1. Next to Create Monitoring User, click No.</li><li data-bbox="415 842 959 873">2. Enter the user name and password.</li></ol> <p data-bbox="383 909 1511 940">If you create the user manually, DPA uses the default Tablespaces for that user.</p> <p data-bbox="383 972 1511 1131">If you are registering multiple Oracle Real Application Clusters (RAC) nodes, you may receive an error that the user already exists. You can create a different monitoring user or clear the Create a New Monitoring User check box and continue.</p>


Panel	Instructions
Oracle Monitoring Information	<p data-bbox="373 216 1523 300">If the monitored instance contains the Oracle E-Business Suite, DPA can collect additional information about the suite.</p> <p data-bbox="373 321 1523 489">DPA can capture Oracle E-Business data to identify the screens, modules, and users generating the database requests. This gives you increased visibility into the causes of performance problems in the Oracle E-Business Suite, Oracle Enterprise Resource Planning (ERP), and Oracle Applications environments.</p> <p data-bbox="373 510 1523 636">The SYS password is requested only if remote login as SYS is enabled on the monitored Oracle instance. This option is not available for Amazon RDS instances.</p> <p data-bbox="373 657 1523 741">If you do not have remote SYS access to the computer, click the link to open the Manual Steps for Monitored Database Instance Registration.</p> <p data-bbox="373 762 1523 888">You can run a script to run on the monitored instance to install a utility package for DPA that grants Execute permissions for that package to the monitoring user.</p> <p data-bbox="373 909 1523 951">To register the monitored database instance manually:</p> <ol data-bbox="406 972 1523 1339" style="list-style-type: none"><li data-bbox="406 972 1523 1014">1. Click Register the monitored database instance manually.</li><li data-bbox="406 1035 1523 1077">2. Click Select All, copy the script, and paste it into a text file.</li><li data-bbox="406 1098 1523 1182">3. As an Oracle Administrator, log in as SYS to the database instance to be monitored.</li><li data-bbox="406 1203 1523 1245">4. Access the text file.</li><li data-bbox="406 1266 1523 1339">5. Execute the script.</li></ol>
Oracle Repository Tablespace	<p data-bbox="373 1350 1523 1392">If your repository database is not Oracle, the wizard skips this step.</p> <p data-bbox="373 1413 1523 1497">Choose the tablespace in the repository database to store DPA performance data for this monitored instance.</p> <p data-bbox="373 1518 1523 1642">By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.</p>

Panel	Instructions
Select the Alert Groups	<p>If you have no Alert Groups set up, or if this new database instance does not match the database type of the Alert Group, the wizard skips this step.</p> <p>Alert Groups simplify alert configuration and help make alerting more consistent across the monitored database instances.</p> <p>Select the Alert Groups you want the new database instance to join.</p>
Summary	Review the information and click Register Database Instance.
Database Instance Registration Complete	Click Finish to return to the DPA homepage.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register a SQL Server database

Complete the following tasks to register an individual SQL Server database instance for monitoring with DPA.

-  If you are monitoring a SQL Server Availability Group (AG), see [Registration and licensing options for clustered environments](#).
- You can use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

## Identify the privileged user



When you register a database instance, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For self-managed SQL Server database instances, the privileged user requires the following privileges:

`SYSADMIN` role

## Complete the registration wizard

1. On the DPA homepage, click Register DB Instance for Monitoring.
2. Under Self-Managed, click Microsoft SQL Server.
3. Click Next.
4. Complete the wizard panels as described in the following table.

Panel	Instructions
Enter Monitored Database Instance Connection Information	<ol style="list-style-type: none"><li data-bbox="508 216 1482 625">1. Enter connection information for the SQL Server instance:<ul style="list-style-type: none"><li data-bbox="602 279 1482 405">• If the SQL Server Browser service is running, enter the server name or IP address and the instance name in this format: <code>Server\Instance</code>.</li><li data-bbox="602 415 1482 541">• If the SQL Server instance contains one or more Availability Groups, click Note for Availability Groups for instructions on how to register primary and secondary replicas.</li><li data-bbox="602 552 1482 625">• Otherwise, enter the server name or IP address and the port number.</li></ul><div data-bbox="557 653 1515 800" style="border: 1px solid #ccc; padding: 5px;"><p> DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.</p></div><ol style="list-style-type: none"><li data-bbox="508 825 1482 940">2. Select the type of authentication you want to use. If Mixed Mode was selected during the SQL Server installation, you can choose either option.</li><li data-bbox="508 972 1482 1003">3. Enter a SYSADMIN login that DPA can use to register the instance.<div data-bbox="557 1035 1515 1140" style="border: 1px solid #ccc; padding: 5px;"><p> DPA does not use or store these credentials after you complete the wizard.</p></div><ul style="list-style-type: none"><li data-bbox="602 1161 1482 1245">• For Windows authentication, enter <code>&lt;DOMAIN&gt;\&lt;username&gt;</code> in the SYSADMIN User field.</li><li data-bbox="602 1266 1482 1434">• For SQL Server authentication, enter the credentials that you enter on the Connect to Server dialog in SQL Server Management Studio (with Database Engine as the Server type).</li></ul></li></ol><p data-bbox="475 1455 1482 1539">SSL is requested by default. If the server does not support SSL, a plain connection is used.</p><p data-bbox="475 1560 1336 1642">Are you receiving errors? See <a href="#">DPA for SQL Server installation troubleshooting</a>.</p></li></ol>

Panel	Instructions
Enter the Monitoring User	<p>Create or specify the account that DPA will use to gather information. To ensure that the account has the required permissions, SolarWinds recommends creating a new account.</p> <p>To create a new account:</p> <ol style="list-style-type: none"> <li>1. Click Yes.</li> <li>2. Select SQL Server as the authentication method. (DPA cannot create a new Windows account.)</li> <li>3. Enter a user name and password for the new account, or accept the default values.</li> </ol> <p>To specify an existing account:</p> <ol style="list-style-type: none"> <li>1. Click No.</li> <li>2. Select either authentication method.</li> <li>3. Enter the user name and password of an existing account.</li> </ol> <p>For Windows authentication, enter <code>&lt;DOMAIN&gt;\&lt;username&gt;</code> in the Monitoring User field.</p> <p>You can also authenticate <a href="#">using a Windows Computer Account</a>.</p> <p>For SQL Server authentication, only the user name is required. Do not specify a domain.</p>
Oracle Repository Tablespace	<p>If your repository database is not Oracle, the wizard skips this step.</p> <p>Choose the tablespace in the repository database to store DPA performance data for this monitored instance.</p> <p>By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.</p>
Select the Alert Groups	<p>If you have no Alert Groups set up, or if this new database instance does not match the database type of the Alert Group, the wizard skips this step.</p> <p>Alert Groups simplify alert configuration and help make alerting more consistent across the monitored database instances.</p> <p>Select the Alert Groups you want the new database instance to join.</p>



Panel	Instructions
Summary	Review the information and click Register Database Instance.
Database Instance Registration Complete	Click Finish to return to the DPA homepage.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register a Sybase database

Complete the following steps to register an individual Sybase database instance for monitoring with DPA.

You can also use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

### Identify the privileged user

When you register a database instance, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For Sybase database instances, the privileged user requires the following privileges:

SA\_ROLE

### Complete the registration wizard

1. On the DPA homepage, click Register DB Instance for Monitoring.
2. Under Self-Managed, click SAP Sybase ASE.
3. Click Next.
4. Complete the wizard panels as described in the following table.

Panel	Instructions
Enter Monitored Database Instance Connection Information	<p>Enter the server name or IP address and port of the Sybase server.</p> <p>DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.</p> <p>Enter SA_ROLE credentials for DPA to register the database instance.</p> <p>The Sybase Monitor Server does not need to be configured for DPA to monitor the database.</p>
Enter the Monitoring User	<p>Create or specify the account that DPA will use to gather information. To ensure that the account has the required permissions, SolarWinds recommends creating a new account.</p> <p>To create a new account:</p> <ol style="list-style-type: none"><li>1. Next to Create Monitoring User, click Yes.</li><li>2. Select the Authentication method, and enter the user name and password.</li></ol> <p>To specify an existing account:</p> <ol style="list-style-type: none"><li>1. Next to Create Monitoring User, click No.</li><li>2. Enter the user name and password.</li></ol> <p>DPA requires the monitoring user to have SA_ROLE and MON_ROLE privileges for data collection.</p> <p>DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will not cause load on the monitored instance.</p>
Oracle Repository Tablespace	<p>If your repository database is not Oracle, the wizard skips this step.</p> <p>Choose the tablespace in the repository database to store DPA performance data for this monitored instance.</p> <p>By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.</p>

Panel	Instructions
Select the Alert Groups	<p>If you have no Alert Groups set up, or if this new database instance does not match the database type of the Alert Group, the wizard skips this step.</p> <p>Alert Groups simplify alert configuration and help make alerting more consistent across the monitored database instances.</p> <p>Select the Alert Groups you want the new database instance to join.</p>
Summary	Review the information and click Register Database Instance.
Database Instance Registration Complete	Click Finish to return to the DPA homepage.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register a Db2 database instance

Complete the following tasks to register an individual Db2 database instance for monitoring with DPA.

You can also use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

### Identify the privileged user

When you register a database instance, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For Db2 database instances, the privileged user requires the following privileges:

SYSADM

### Complete the registration wizard

1. On the DPA homepage, click Register DB Instance for Monitoring.
2. Under Self-Managed, click DB2 UDB.

3. Click Next.
4. Complete the wizard panels as described in the following table.

Panel	Instructions
Db2 Configuration Settings	<p>DPA requires the Db2 instance-wide parameter <code>{DFT_MON_STMT}</code> to be turned on to collect monitoring data. Follow the on-screen instructions to check and set the parameter.</p> <p>If <code>{DFT_MON_STMT}</code> is set to <code>OFF</code>, you can use DPA to register the database instance. Later, you can set it to <code>ON</code> and restart the database instance during an approved maintenance window. In the meantime, the database shows a status of <code>Idle</code>.</p>
Enter Monitored Database Instance Connection Information	<p>Enter the host name or IP address and port of the Db2 server.</p> <p>Enter the Db2 database for DPA to monitor in the Database field. DPA collects information from all Db2 instances in a cluster configuration for the specified database.</p> <p>If the connection information changes for the Db2 server, all databases on that instance must be updated separately through the Update Database Instance Connection Wizard.</p> <p>DPA can monitor all databases in the specified instance, or an individual database.</p> <ul style="list-style-type: none"> <li>• DPA 9.0 and later monitors all Db2 databases in the specified instance.</li> <li>• Do you want to monitor a single database in an earlier version of DPA? See <a href="#">Switch to Db2 instance-wide monitoring</a>.</li> <li>• To monitor a single database, each database must be registered separately through this wizard, even if multiple databases are contained on a single Db2 server instance.</li> <li>• For instance-wide monitoring, one database must be registered for the DPA connection.</li> </ul> <p>Enter SYSADM credentials for DPA to monitor the database instance.</p> <p>Do you want more information on the Db2 permissions needed by DPA for the monitoring user? See <a href="#">Required Db2 permissions needed by DPA for monitoring</a>.</p>

Panel	Instructions
Enter the Monitoring User	<p>Create or specify the account that DPA will use to gather information. To ensure that the account has the required permissions, SolarWinds recommends creating a new account.</p> <p>To create a new account:</p> <ol style="list-style-type: none"><li>1. Next to Create Monitoring User, click Yes.</li><li>2. Select the Authentication method, and enter the user name and password.</li></ol> <p>To specify an existing account:</p> <ol style="list-style-type: none"><li>1. Next to Create Monitoring User, click No.</li><li>2. Enter the user name and password.</li></ol> <p>DPA requires the monitoring user to have SA_ROLE and MON_ROLE privileges for data collection.</p> <p>DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will not cause load on the monitored instance.</p>
Oracle Repository Tablespace	<p>If your repository database is not Oracle, the wizard skips this step.</p> <p>Choose the tablespace in the repository database to store DPA performance data for this monitored instance.</p> <p>By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.</p>
Select the Alert Groups	<p>If you have no Alert Groups set up, or if this new database instance does not match the database type of the Alert Group, the wizard skips this step.</p> <p>Alert Groups simplify alert configuration and help make alerting more consistent across the monitored database instances.</p> <p>Select the Alert Groups you want the new database instance to join.</p>
Summary	<p>Review the information and click Register Database Instance.</p>
Database Instance Registration Complete	<p>Click Finish to return to the DPA homepage.</p>

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register a MySQL, Percona, or Maria database

Complete the following steps to register an individual MySQL, Percona, or Maria database instance for monitoring with DPA.

To optimize DPA's reporting capabilities for a MySQL, Percona, or Maria database instance, see the [requirements for monitoring MySQL database instances](#).

You can also use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

### Identify the privileged user

When you register a database instance, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For self-managed MySQL, Percona, or Maria database instances:

- The privileged user requires the following permission:

```
CREATE USER
```

- The privileged user must be able to grant the following permissions:

```
PROCESS on *.*  
SELECT & UPDATE on performance_schema.*
```

- To enable the retrieval of query execution plans, the privileged user must also be able to grant the following permissions:

```
SELECT, INSERT, UPDATE, DELETE on *.*  
SYSADM
```

### Complete the registration wizard

1. On the DPA homepage, click Register DB Instance for Monitoring.
2. Under Self-Managed, click MySQL.
3. Click Next.
4. Complete the wizard panels as described in the following table.

Panel	Instructions
Enter Monitored Database Instance Connection Information	<p data-bbox="378 220 1177 256">Enter the host name or IP address and port of the server.</p> <p data-bbox="378 287 1437 363">DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.</p> <p data-bbox="378 394 1497 554">DPA ignores data generated by the monitoring user on the monitored database instance. For this reason, do not specify a user that causes load on the monitored instance. SolarWinds recommends creating a separate account for the monitoring user.</p> <p data-bbox="378 585 734 621">To create a new account:</p> <ol data-bbox="412 653 1417 791" style="list-style-type: none"><li data-bbox="412 653 883 688">1. Click Provide a privileged user.</li><li data-bbox="412 720 1417 791">2. Enter the credentials of an existing user with privileges to create the monitoring user and to grant the required permissions.</li></ol> <p data-bbox="456 823 1433 898">The credentials for the privileged user are not used or stored after the registration.</p> <p data-bbox="456 930 1458 1005">The privileged user requires the CREATE USER permission and must be able to grant the following permissions:</p> <pre data-bbox="456 1037 1156 1106">PROCESS on *.* SELECT &amp; UPDATE on performance_schema.*</pre> <p data-bbox="456 1138 1487 1213">To enable the retrieval of query execution plans, this privileged user must be able to grant the following permissions:</p> <pre data-bbox="456 1245 1118 1272">SELECT, INSERT, UPDATE, DELETE on *.*</pre> <ol data-bbox="412 1304 1510 1379" style="list-style-type: none"><li data-bbox="412 1304 1510 1379">3. Enter credentials for the monitoring user. You can create a new user or use an existing one.</li></ol> <p data-bbox="378 1411 820 1446">To specify an existing account:</p> <ol data-bbox="412 1478 1109 1577" style="list-style-type: none"><li data-bbox="412 1478 932 1514">1. Click Provide the monitoring user.</li><li data-bbox="412 1545 1109 1577">2. Enter credentials. DPA encrypts the password.</li></ol> <p data-bbox="378 1608 1469 1684">Alternatively, you can use the script that DPA provides to create a monitoring user.</p> <ol data-bbox="412 1715 1341 1915" style="list-style-type: none"><li data-bbox="412 1715 1341 1791">1. Click Monitoring User Creation Script, and follow the on-screen instructions.</li><li data-bbox="412 1822 1243 1858">2. Copy the edited script to the MySQL console, and run it.</li><li data-bbox="412 1887 1044 1915">3. Provide this user as your monitoring user.</li></ol>

Panel	Instructions
Oracle Repository Tablespace	<p>If your repository database is not Oracle, the wizard skips this step.</p> <p>Choose the tablespace in the repository database to store DPA performance data for this monitored instance.</p> <p>By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.</p>
Select the Alert Groups	<p>If you have no Alert Groups set up, or if this new database instance does not match the database type of the Alert Group, the wizard skips this step.</p> <p>Alert Groups simplify alert configuration and help make alerting more consistent across the monitored database instances.</p> <p>Select the Alert Groups you want the new database instance to join.</p>



**Panel Instructions**

MySQL Configuration for Monitoring

Select a Typical or Custom configuration. SolarWinds recommends the Typical configuration.

- The DPA Recommended option is used for Performance Schema setup.
- `EXPLAIN` can be run on `SELECT` statements.

Select Custom to change the Performance Schema setup and to allow `EXPLAIN` to be run on different statements.

Performance Schema setup

Specify what data the Performance Schema collects and maintains. This table shows which consumers and instruments each option enables.

**i** The MySQL Performance Schema must be enabled. If you select Leave As Is, verify that Global Instrumentation and Thread Instrumentation are enabled in the existing Performance Schema configuration.

Option	Server Default	DPA Recommended	Detailed	Leave as Is
Consumer Global Instrumentation	✓	✓	✓	NC
Consumer Thread Instrumentation	✓	✓	✓	NC
Consumer Statement Digest	✓	✓	✓	NC
Consumer Statement (Current)	✓	✓	✓	NC
Consumer Wait (Current)		✓	✓	NC
Instrument Wait (Lock/*)		✓	✓	NC
Instrument Wait (I/O table) (I/O/file)		✓	✓	NC
Instrument Wait (I/O/socket)		✓	✓	NC

Panel	Instructions				
	Option	Server Default	DPA Recommended	Detailed	Leave as Is
	Instrument Wait (Synch/*)			✓	NC

### MySQL Configuration for Monitoring (continued)

✓ = Enabled.

NC = No change. DPA does not change the existing Performance Schema configuration.

\* Values that are outside of the `MYSQL_PERFORMANCE_SCHEMA` configuration scope of DPA are not changed. For example, an instrument named `stage` exists in the MySQL Performance Schema. If you enable or disable that instrument, DPA will not change it.

Allow EXPLAIN to be run on

This section is displayed if you specified a privileged user to create the DPA monitoring user.

Select what type of statements you want DPA to collect execution plans for. The monitoring user can run `EXPLAIN` on the selected statement types.

### Summary

Review the information and click Register Database Instance.


### Database Instance Registration Complete

Click Finish to return to the DPA homepage.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register a PostgreSQL database instance and prepare for monitoring

Complete the following tasks to register a PostgreSQL database instance for monitoring with DPA.

 You can also use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

Registering a PostgreSQL database instance is slightly different than registering other types of monitored database instances:

- You cannot use the wizard to create the DPA monitoring user. Create the monitoring user manually, as described below.
- If the DPA repository is an Oracle database, DPA stores performance data for monitored PostgreSQL database instances in the default tablespace of the repository user. You cannot change the default tablespace in the Register Instance Wizard. If you need to change the default tablespace, register the instance using [mass registration](#).
- You must configure each PostgreSQL database instance, as described below.

## Task 1: Create the DPA monitoring user

Use these instructions to manually create the user that DPA uses to monitor a PostgreSQL database instance. The user will have the necessary rights and privileges.

DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will not cause load on the monitored instance.

1. Run the following SQL statement on the PostgreSQL database instance to create the DPA monitoring user:

```
CREATE USER dpa_user WITH ENCRYPTED PASSWORD 'password';
```

where *dpa\_user* is the user name and *password* is the password.

2. Grant privileges to the user.

**i** There are dedicated `pg_read_all_stats` and `pg_read_all_settings` roles in PostgreSQL 10 and later. For earlier versions, the `SUPERUSER` privilege is required.

- For PostgreSQL 10.x and later:

```
GRANT pg_read_all_stats, pg_read_all_settings, pg_signal_backend TO dpa_user;
```

- For PostgreSQL 9.6.x in on-premises deployments:

```
ALTER USER dpa_user WITH SUPERUSER;
```

- For PostgreSQL 9.6.x in Amazon RDS and Amazon Aurora deployments:

```
GRANT rds_superuser TO dpa_user;
```

- For PostgreSQL 9.6.x in Azure deployments:

```
GRANT azure_pg_admin TO dpa_user;
```

3. If you are monitoring EDB Postgres version 10, you must give the DPA monitoring user access to the `pg_stat_statements` view.

**i** For EDB Postgres version 10, granting the the `pg_read_all_stats` role does **not** give the DPA monitoring user access to the view `pg_stat_statements`.

To grant access, create a dedicated DPA schema and make a synonym of `pg_stat_statements` in it:

```
CREATE SCHEMA dpa_schema;  
CREATE VIEW dpa_schema.pg_stat_statements AS SELECT * FROM enterprisedb.pg_  
stat_statements;  
GRANT USAGE ON SCHEMA dpa_schema TO dpa_user  
GRANT SELECT ON dpa_schema.pg_stat_statements TO dpa_user;
```

## Task 2: Configure PostgreSQL database instances for DPA monitoring

### Determine which monitoring mode to use

DPA offers two modes of monitoring PostgreSQL database instances. The monitoring mode you choose determines what configuration steps are required.

- **Limited monitoring** queries only the `pg_stat_activity` view. The `pg_stat_activity` view is a system view containing information about database server processes activity. Limited monitoring:
  - Is sufficient for getting wait time information for queries.
  - Returns incomplete SQL texts, and query execution statistics might be missing.
- **Complete monitoring** queries both the `pg_stat_activity` and `pg_stat_statements` views. The `pg_stat_statements` view contains execution statistics for all SQL statements executed by a server. Complete monitoring:
  - Provides complete SQL texts and query execution statistics.
  - Requires additional `pg_stat_statements` extension configuration (described in the following section).

**i** PostgreSQL is delivered as a set of mandatory and optional packages. The `pg_stat_statements` extension provides a means for tracking SQL statement execution statistics and is required for complete monitoring. This extension is included by default in PostgreSQL distributions for Linux and Windows OS. Installation of other extensions is platform-dependent. See <https://www.postgresql.org/download/> for more information.

## Configure each database instance

Complete the following steps to configure each PostgreSQL database instance that you want to monitor.

### 1. Enable remote access to the PostgreSQL instance.

**i** Remote access is enabled by default for EDB Standard and EDB Enterprise editions. For those versions, you can skip step 1b below.

- a. Adjust firewall rules to allow an incoming connection from DPA to the monitored instance. Ensure that the port the PostgreSQL instances is listening on is open (port 5432 by default).
- b. (For editions other than EDB Standard and EDB Enterprise) To configure PostgreSQL accessibility, edit the `postgresql.conf` configuration file and change the `listen_address` property value to:

```
listen_address = '*'
```

**i** Alternatively, you can append the IP address of the DPA server to a comma-separated list of addresses.

- c. To configure authentication methods for the DPA user, edit the `pg_hba.conf` configuration file and add the following host record:

```
host all dpa_user all md5
```

where `dpa_user` is the DPA monitoring user name [created previously](#).

- d. If the `pg_hba.conf` configuration file restricts access to the monitored instance to a range of IP addresses, ensure that the DPA server is included in the IP address range.
- e. Restart the PostgreSQL server.

### 2. If you want to perform [complete monitoring](#), enable and configure the `pg_stat_statements` extension for Text Poll and Stats Poll functionality:

- a. Run the following command to determine if the extension is installed:

```
SELECT * FROM pg_available_extensions WHERE name = 'pg_stat_statements';
```

If there is no installed version or you receive the error `pg_stat_statements does not exist`, you must load the extension (as described in the following step). The extension is loaded by adding `pg_stat_statements` entry to `shared_preload_libraries` because it requires additional shared memory.

b. To load the `pg_stat_statements` extension (if needed) and configure it, perform one of the following tasks:

- For on-premises deployments, edit the `postgresql.conf` file and add or modify the following entries:

```
shared_preload_libraries = 'pg_stat_statements'
track_activity_query_size = 2048
pg_stat_statements.track = top
```

**i** Optionally, you can enter `pg_stat_statements.track = all` instead of `pg_stat_statements.track = top`.

- For Amazon RDS deployments, use the AWS Console to modify your existing custom DB Parameter Group or create a new DB Parameter Group. Then enter the following parameter values:

Parameter name	Value
<code>pg_stat_statements.track</code>	ALL
<code>shared_preload_libraries</code>	<code>pg_stat_statements</code>
<code>track_activity_query_size</code>	2048

- For Azure deployments, modify your Server parameters to include the parameter values listed in the previous table.

c. Restart the PostgreSQL server.

3. Create the `pg_stat_statements` extension in the database. The extension is database-bound and must be created for each database.

**i** The `pg_stat_statements` extension must be created in the database used to connect to DPA.

To create the extension:

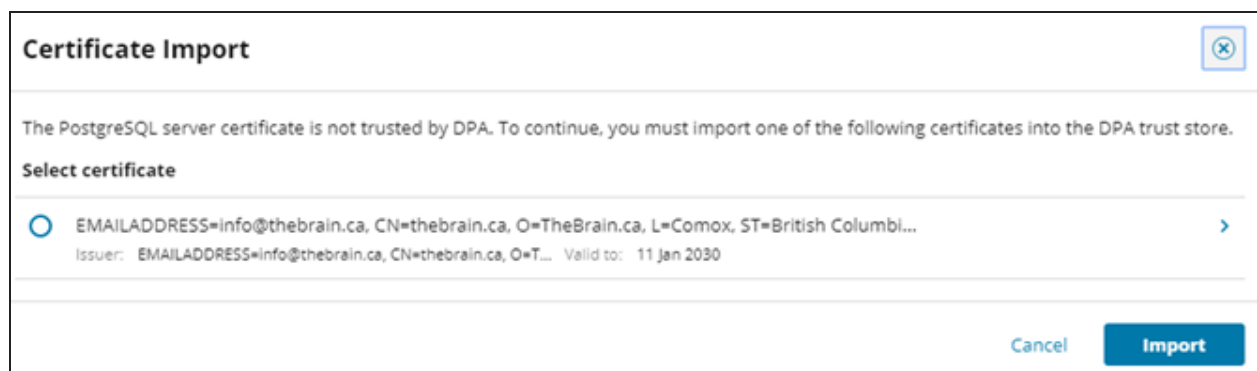
- Connect to the PostgreSQL database instance with the DPA user account or superuser (for EDB Enterprise edition).
- Execute following command:

```
CREATE EXTENSION pg_stat_statements;
```

## Task 3: Run the Register Instance Wizard

1. In the upper-left corner of the DPA homepage, click Register DB Instance for Monitoring.
2. Under Self-Managed, Amazon, or Azure, click PostgreSQL.
3. Click Next.
4. Complete the Connection Information panel:
  - a. Enter the server name or IP address of the database instance and the port number.
  - b. Under SSL mode, specify the type of secure socket layer (SSL) connections established between the instance and the DPA server:
    - Disable: SSL connections are not used.
    - Require: SSL is enabled, but no certificate checks are performed.
    - Verify-CA: SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA).

If you select this option and DPA cannot access a trusted certificate, you are prompted to import a certificate into the [DPA trust store](#). Click the arrow on the right to view certificate details.



- Verify-Full: SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN).

If you select this option and DPA cannot access a trusted certificate, you are prompted to import a certificate into the [DPA trust store](#). Click the arrow on the right to view certificate details.
- c. Select the authentication method used when the DPA monitoring user connects to this database instance.
  - d. Enter the user name and password for the monitoring user account that you [created](#)

[previously](#).

e. Click Next.

5. Complete the Instance Options panel:

a. Enter the name that DPA will display to identify this database instance.

The Display name field defaults to the name retrieved from the database instance.

b. (Optional) If you have existing [database instance groups](#), you can assign this database instance to one of the groups.

**i** If you do not have database instance groups, the Instance group field is not shown.

c. (Optional) If you have existing [alert groups](#), you can assign this database instance to one or more groups.

**i** If you do not have alert groups, the Alert group field is not shown.

d. Click Next.

6. Review the Summary panel:

a. Review the information. If necessary, click Back to make any corrections.

b. When all information is correct, click Register.

## Register an Amazon RDS for Oracle database

Complete the following steps to register an individual Amazon RDS for Oracle database instance for monitoring with DPA.

You can also use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

1. On the DPA homepage, click Register DB Instance for Monitoring.


2. Under Amazon RDS, click Amazon RDS for Oracle.


3. Click Next.

4. Complete the wizard panels as described in the following table.

**i** If registration fails because your DPA server cannot connect to the instance's server, see [DPA database registration failure when attempting to register a database on an external network](#).



Panel	Instructions
Enter Monitored Database Instance Connection Information	<p>Amazon Relational Database Service (RDS) for Oracle database instances have three connection options:</p> <ul style="list-style-type: none"><li>• Direct Connect</li><li>• Transparent Network Substrate (TNS) Connect Descriptor</li><li>• Lightweight Directory Access Protocol (LDAP) or TNS Name</li></ul> <p>Direct Connect</p> <p>Enter the Service Name or System Identifier (SID), host name or IP address, and port. The default port is 1521.</p> <p>TNS Connect Descriptor</p> <p>The Connect Descriptor value contains everything after <code>NAME=</code> in the <code>tnsnames.ora</code> file. The beginning <code>(DESCRIPTION=</code> is necessary. For example:</p> <pre>(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST = demo.confio.com) (PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = demo)))</pre> <p>LDAP or TNS Name</p> <p>To use this option, Oracle Name Resolution must be configured. For instructions, see <a href="#">Connect to Oracle using name resolution</a>.</p> <p>After you configure Oracle Name Resolution, you can use the LDAP/TNS Name when registering additional monitored database instances.</p> <p>RAC instances</p> <p>For an Oracle RAC (Real Application Cluster), register every physical instance in the cluster. Do not register the virtual IP that distributes load across the RAC instances.</p> <div data-bbox="345 1514 1513 1654" style="border: 1px solid #ccc; padding: 10px;"><p> If you choose to register the virtual IP load balancing listener, or to monitor only a subset of instances in the cluster, DPA will not have complete and consistent data. This will affect DPA's tuning and resource analysis.</p></div> <p>For more information, see <a href="#">DPA registration and licensing options for clustered environments</a>.</p> <p>DBA user</p> <p>Enter DBA credentials for DPA to register the database instance.</p>

Panel	Instructions
Enter the Monitoring User	<p>Create or specify the account that DPA will use to gather information. To ensure that the account has the required permissions, SolarWinds recommends creating a new account.</p> <p>To create a new account:</p> <ol style="list-style-type: none"> <li>1. Next to Create Monitoring User, click Yes.</li> <li>2. Enter the user name and password.</li> <li>3. Select a Tablespace and Temp Tablespace on the monitored database. This is primarily used for gathering Explain Plan data for monitored queries.</li> </ol> <p>To specify an existing account:</p> <ol style="list-style-type: none"> <li>1. Next to Create Monitoring User, click No.</li> <li>2. Enter the user name and password.</li> </ol> <p>If you create the user manually, DPA uses the default Tablespaces for that user.</p>
Oracle Monitoring Information	<p>If the monitored instance contains the Oracle E-Business Suite, DPA can collect additional information about the suite.</p> <p>DPA can capture Oracle E-Business data to identify the screens, modules, and users generating the database requests. This gives you increased visibility into the causes of performance problems in the Oracle E-Business Suite, Oracle Enterprise Resource Planning (ERP), and Oracle Applications environments.</p> <div data-bbox="345 1230 1515 1291" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> The SYS password option is not available for Amazon RDS instances.</p> </div>
Oracle Repository Tablespace	<p>If your repository database is not Oracle, the wizard skips this step.</p> <p>Choose the tablespace in the repository database to store DPA performance data for this monitored instance.</p> <p>By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.</p>
Select the Alert Groups	<p>If you have no Alert Groups set up, or if this new database instance does not match the database type of the Alert Group, the wizard skips this step.</p> <p>Alert Groups simplify alert configuration and help make alerting more consistent across the monitored database instances.</p> <p>Select the Alert Groups you want the new database instance to join.</p>

Panel	Instructions
Summary	Review the information and click Register Database Instance.
Database Instance Registration Complete	Click Finish to return to the DPA homepage.


If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register an Amazon RDS for SQL Server database

Complete the following steps to register an individual Amazon RDS for SQL Server database instance for monitoring with DPA.

You can also use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

1. On the DPA homepage, click Register DB Instance for Monitoring.
2. Under Amazon RDS, click Amazon RDS for SQL Server.
3. Click Next.
4. Complete the wizard panels as described in the following sections.

 If registration fails because your DPA server cannot connect to the instance's server, see [DPA database registration failure when attempting to register a database on an external network](#).

## Connection Information

Enter the server name or IP address and port.

DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.

SSL is requested by default. If the server does not support SSL, a plain connection is used.

Enter an Amazon RDS Master User for DPA to register the database instance. If you do not want to enter the Master User that created the database instance, use the SQL statement below to create a new Master User. Replace `dpa` with the new user name.

```
CREATE LOGIN [dpa] WITH PASSWORD=N'Password1';
        GRANT ALTER ANY LOGIN TO dpa;
        GRANT VIEW SERVER STATE TO dpa WITH GRANT OPTION;
        GRANT VIEW ANY DEFINITION TO dpa WITH GRANT OPTION;
        GRANT VIEW ANY DATABASE TO dpa WITH GRANT OPTION;
ALTER SERVER ROLE [processadmin] ADD MEMBER [dpa];
```

**Panel****Instructions****Enter the Monitoring User**

Create or specify the account that DPA will use to gather information. To ensure that the account has the required permissions, SolarWinds recommends creating a new account.

To create a new account:

1. Click Yes.
2. Select SQL Server as the authentication method. (DPA cannot create a new Windows account.)
3. Enter a user name and password for the new account, or accept the default values.

To specify an existing account:

1. Click No.
2. Select either authentication method.
3. Enter the user name and password of an existing account.

For Windows authentication, enter `<DOMAIN>\<username>` in the Monitoring User field.

You can also authenticate [using a Windows Computer Account](#).

For SQL Server authentication, only the user name is required. Do not specify a domain.

Panel	Instructions
Oracle Repository Tablespace	<p>If your repository database is not Oracle, the wizard skips this step.</p> <p>Choose the tablespace in the repository database to store DPA performance data for this monitored instance.</p> <p>By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.</p>
Select the Alert Groups	<p>If you have no Alert Groups set up, or if this new database instance does not match the database type of the Alert Group, the wizard skips this step.</p> <p>Alert Groups simplify alert configuration and help make alerting more consistent across the monitored database instances.</p> <p>Select the Alert Groups you want the new database instance to join.</p>
Summary	Review the information and click Register Database Instance.
Database Instance Registration Complete	Click Finish to return to the DPA homepage.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register an Amazon RDS for MySQL or Aurora database instance

Complete one of the following tasks to register an individual Amazon RDS for MySQL or Amazon Aurora database instance for monitoring with DPA.

You can also use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

### Register a read-only Amazon RDS for MySQL database instance

To register a **read-only** Amazon RDS for MySQL database instance, complete the following steps:

1. Register the corresponding read/write instance in DPA using the [registration wizard](#).
2. Copy the user and permissions to the read-only instance.

3. Open the following file in a text editor:

```
<DPA_Home>\iwc\tomcat\ignite_config\idc\system.properties
```

4. Add the following setting to the system.properties file and save it:

```
com.confio.idc.wizard.allowDuplicateDatabaseRegistration=true
```

5. Use the [registration wizard](#) to register the instance:

- For the monitoring user, choose Provide monitoring user. Then enter the credentials for the same user specified for the read/write instance.
- On the Configuration for Monitoring panel, choose Leave As Is.

## Run the registration wizard

To register an Amazon RDS for MySQL or Aurora database for DPA to monitor:

1. On the DPA homepage, click Register DB Instance for Monitoring.
2. Under Amazon RDS, click Amazon RDS for MySQL.
3. Click Next.
4. Complete the remaining wizard panels as described in the following table.

**i** If registration fails because your DPA server cannot connect to the instance's server, see [DPA database registration failure when attempting to register a database on an external network](#).

Panel	Instructions
Enter Monitored Database Instance Connection Information	<p data-bbox="378 222 1177 258">Enter the host name or IP address and port of the server.</p> <p data-bbox="378 289 1437 363">DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.</p> <p data-bbox="378 394 1497 552">DPA ignores data generated by the monitoring user on the monitored database instance. For this reason, do not specify a user that causes load on the monitored instance. SolarWinds recommends creating a separate account for the monitoring user.</p> <p data-bbox="378 583 734 619">To create a new account:</p> <ol data-bbox="412 651 1417 793" style="list-style-type: none"><li data-bbox="412 651 885 686">1. Click Provide a privileged user.</li><li data-bbox="412 718 1417 793">2. Enter the credentials of an existing user with privileges to create the monitoring user and to grant the required permissions.</li></ol> <p data-bbox="456 825 1433 898">The credentials for the privileged user are not used or stored after the registration.</p> <p data-bbox="456 930 1458 1003">The privileged user requires the CREATE USER permission and must be able to grant the following permissions:</p> <pre data-bbox="456 1035 1156 1108">PROCESS on *.* SELECT &amp; UPDATE on performance_schema.*</pre> <p data-bbox="456 1140 1485 1213">To enable the retrieval of query execution plans, this privileged user must be able to grant the following permissions:</p> <pre data-bbox="456 1245 1117 1276">SELECT, INSERT, UPDATE, DELETE on *.*</pre> <ol data-bbox="412 1308 1510 1381" style="list-style-type: none"><li data-bbox="412 1308 1510 1381">3. Enter credentials for the monitoring user. You can create a new user or use an existing one.</li></ol> <p data-bbox="378 1413 820 1449">To specify an existing account:</p> <ol data-bbox="412 1480 1109 1581" style="list-style-type: none"><li data-bbox="412 1480 933 1516">1. Click Provide the monitoring user.</li><li data-bbox="412 1547 1109 1581">2. Enter credentials. DPA encrypts the password.</li></ol> <p data-bbox="378 1612 1469 1686">Alternatively, you can use the script that DPA provides to create a monitoring user.</p> <ol data-bbox="412 1717 1339 1915" style="list-style-type: none"><li data-bbox="412 1717 1339 1791">1. Click Monitoring User Creation Script, and follow the on-screen instructions.</li><li data-bbox="412 1822 1242 1858">2. Copy the edited script to the MySQL console, and run it.</li><li data-bbox="412 1887 1044 1915">3. Provide this user as your monitoring user.</li></ol>

Panel	Instructions
Oracle Repository Tablespace	<p>If your repository database is not Oracle, the wizard skips this step.</p> <p>Choose the tablespace in the repository database to store DPA performance data for this monitored instance.</p> <p>By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.</p>
Select the Alert Groups	<p>If you have no Alert Groups set up, or if this new database instance does not match the database type of the Alert Group, the wizard skips this step.</p> <p>Alert Groups simplify alert configuration and help make alerting more consistent across the monitored database instances.</p> <p>Select the Alert Groups you want the new database instance to join.</p>



**Panel Instructions**

**MySQL Configuration for Monitoring**

Select a Typical or Custom configuration. SolarWinds recommends the Typical configuration.

The DPA Recommended option is used for Performance Schema setup. Select Custom to change the Performance Schema setup.

Performance Schema setup

Specify what data the Performance Schema collects and maintains. This table shows which consumers and instruments each option enables.

**i** The MySQL Performance Schema must be enabled. If you select Leave As Is, verify that Global Instrumentation and Thread Instrumentation are enabled in the existing Performance Schema configuration.

Option	Server Default	DPA Recommended	Detailed	Leave as Is
Consumer Global Instrumentation	✓	✓	✓	NC
Consumer Thread Instrumentation	✓	✓	✓	NC
Consumer Statement Digest	✓	✓	✓	NC
Consumer Statement (Current)	✓	✓	✓	NC
Consumer Wait (Current)		✓	✓	NC
Instrument Wait (Lock/*)		✓	✓	NC
Instrument Wait (I/O table) (I/O/file)		✓	✓	NC
Instrument Wait (I/O/socket)		✓	✓	NC
Instrument Wait (Synch/*)			✓	NC

Panel	Instructions
MySQL Configuration for Monitoring (continued)	<p>✓ = Enabled.</p> <p>NC = No change. DPA does not change the existing Performance Schema configuration.</p> <p>* Values that are outside of the MYSQL_PERFORMANCE_SCHEMA configuration scope of DPA are not changed. For example, an instrument named <code>stage</code> exists in the MySQL Performance Schema. If you enable or disable that instrument, DPA will not change it.</p>
Summary	Review the information and click Register Database Instance.
Database Instance Registration Complete	Click Finish to return to the DPA homepage.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register an Azure SQL database

Complete the following steps to register an individual Azure SQL database instance for monitoring with DPA.

**i** To register multiple Azure SQL databases using the Mass Registration feature, follow the instructions in [this KB article](#). You can also register database instances using scripts that call the [DPA API](#).

1. On the DPA homepage, click Register DB Instance for Monitoring.
2. Under Azure, click Azure SQL DB.
3. Click Next.
4. Complete the wizard panels as described in the following table.

Panel	Instructions
Enter Monitored Database Instance Connection Information	<p data-bbox="462 220 1461 294">Enter the server name, port, and database name. You cannot use an IP address in the Server Name field.</p> <p data-bbox="462 325 1380 367">Choose a method for creating or configuring the monitoring user.</p> <p data-bbox="462 388 812 430">To create a new account:</p> <ol data-bbox="495 451 1502 640" style="list-style-type: none"><li data-bbox="495 451 1502 535">1. Click Let DPA create a new contained user or configure an existing contained user for me.</li><li data-bbox="495 556 1502 640">2. Enter the credentials of an existing user with privileges to create the monitoring user and to grant the required permissions.</li></ol> <p data-bbox="544 661 1380 703">The privileged user must be a member of the db_owner role.</p> <p data-bbox="544 724 1461 808">The credentials for the privileged user are not used or stored after the registration.</p> <p data-bbox="462 829 893 871">To specify an existing account:</p> <ol data-bbox="495 892 1193 997" style="list-style-type: none"><li data-bbox="495 892 1015 934">1. Click I'll create the database user.</li><li data-bbox="495 955 1193 997">2. Enter credentials. DPA encrypts the password.</li></ol> <p data-bbox="462 1018 1380 1102">Alternatively, you can use the script that DPA provides to create a monitoring user.</p> <ol data-bbox="495 1123 1421 1333" style="list-style-type: none"><li data-bbox="495 1123 1421 1207">1. Click Monitoring User Creation Script, and follow the on-screen instructions.</li><li data-bbox="495 1228 1380 1270">2. Copy and run the edited script on your Azure SQL database.</li><li data-bbox="495 1291 1128 1333">3. Provide this user as your monitoring user.</li></ol>

Panel	Instructions
Enter the Monitoring User	<p>DPA gathers information through this user from the monitored database. You can create a monitoring user through DPA or use an existing user, such as for read-only replica databases.</p> <div data-bbox="467 373 1513 472" style="border: 1px solid #ccc; padding: 5px;"> <p><b>i</b> To register a read-only geo-replica, you must create a monitoring account through the primary server first.</p> </div> <p>SolarWinds recommends creating a new account because DPA requires special permissions that existing users may not have.</p> <p>To create a new account:</p> <ol style="list-style-type: none"> <li>1. Click Let DPA create a new contained user.</li> <li>2. Enter credentials.</li> </ol> <p>To specify an existing account:</p> <ol style="list-style-type: none"> <li>1. Run the following SQL statement on the Azure SQL database: <div data-bbox="544 934 1513 1060" style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <pre>CREATE USER [&lt;USERNAME&gt;] WITH PASSWORD=N'&lt;PASSWORD&gt;'; ALTER ROLE db_owner ADD member &lt;USERNAME&gt;;</pre> </div> </li> <li>2. Click Let DPA configure an existing contained user.</li> <li>3. Enter credentials.</li> </ol>
Oracle Repository Tablespace	<p>If your repository database is not Oracle, the wizard skips this step.</p> <p>Choose the tablespace in the repository database to store DPA performance data for this monitored instance.</p> <p>By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.</p>
Select the Alert Groups	<p>If you have no Alert Groups set up, or if this new database instance does not match the database type of the Alert Group, the wizard skips this step.</p> <p>Alert Groups simplify alert configuration and help make alerting more consistent across the monitored database instances.</p> <p>Select the Alert Groups you want the new database instance to join.</p>
Summary	Review the information and click Register Database Instance.

Panel	Instructions
Database Instance Registration Complete	Click Finish to return to the DPA homepage.

## Enable deadlocks for read-only geo-replicas

To enable the deadlock feature for read-only geo-replica Azure SQL databases, you must create and enable an Extended Event Session (EES).

If you registered the primary server first, an EES is already created and synced. Skip to step 2.

Otherwise, connect to the primary server first to create an EES.

1. Run the following SQL statement:

```
CREATE EVENT SESSION [dpa_deadlock_capture] ON DATABASE
ADD EVENT sqlserver.xml_deadlock_report
ADD TARGET package0.ring_buffer (SET max_events_limit=(1000),
    max_memory=(256))
WITH (MAX_MEMORY = 256KB,
EVENT_RETENTION_MODE = ALLOW_SINGLE_EVENT_LOSS,
MAX_DISPATCH_LATENCY = 30 SECONDS,
MAX_EVENT_SIZE = 0KB,
MEMORY_PARTITION_MODE = NONE,
TRACK_CAUSALITY = OFF,
STARTUP_STATE = ON);
-- ALTER EVENT SESSION [dpa_deadlock_capture] ON DATABASE STATE = START;
```

2. Connect to the read-only replica database.
3. Click Extended Events > Sessions.
4. Enable the dpa\_deadlock\_capture session.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must [activate a license](#) to monitor the database instance.

## Register an Azure SQL Managed Instance

Complete the following tasks to register a single Azure SQL Managed Instance (ASMI) for DPA to monitor.

You can also use mass registration to registrations to [register multiple database instances](#), or you can register database instances using scripts that call the [DPA API](#).

Registering an ASMI is slightly different than registering other types of monitored database instances:

- You cannot use the wizard to create the DPA monitoring user. Create the monitoring user manually, as described below.
- If the DPA repository is an Oracle database, DPA stores performance data for monitored ASMIs in the default tablespace of the repository user. You cannot change the default tablespace in the Register Instance Wizard. If you need to change the default tablespace, register the instance using [mass registration](#).

## Create the monitoring user

In the ASMI, create a user account to serve as the DPA monitoring user. DPA uses this account to register and monitor the instance.

This account must have the SYSADMIN role. For instructions, see [Create the DPA monitoring user for SQL Server and Azure SQL Managed Instance](#).

DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will not cause load on the monitored instance.

## Start the Register Instance Wizard

1. On the DPA homepage, click Register DB Instance for Monitoring.
2. Under Azure, click Azure SQL Managed Instance.
3. Click Next.
4. Complete the wizard panels as described in the following sections.

## Connection Information panel

1. Enter the server name or IP address of the ASMI and the port number.
2. Select the type of authentication you want to use for the monitoring user account.
3. Enter the user name and password for the monitoring user account that you [created previously](#).
4. Click Next.

## Instance Options panel


1. Enter the name that DPA will display to identify this database instance.

The Display name field defaults to the ASMI name retrieved from the instance.

2. (Optional) If you have existing [database instance groups](#), you can assign the ASMI to one of the groups.

 If you do not have database instance groups, the Instance group field is not shown.

3. (Optional) If you have existing [alert groups](#), you can assign the ASMI to one or more groups.

 If you do not have alert groups, the Alert group field is not shown.


4. Click Next.

## Summary panel


1. Review the information. If necessary, click Back to make any corrections.
2. When all information is correct, click Register.

## Unregister a monitored database instance

If you want to remove one of your monitored database instances from DPA, you must unregister it.

 If you unregister a monitored database instance, DPA stops monitoring the instance and removes all historical performance data from the repository.

1. On the DPA menu, click Options.
2. Under Monitor Setup > Database Instances, click Unregister DB Instance.
3. Select a database instance, and click Next.
4. Determine which DPA objects (if any) to remove from the database instance:
  - If the database instance is **not** currently running or cannot be reached, do **not** select any objects. Click Next.


 If you select objects and DPA cannot access the instance to remove them, DPA cannot unregister the instance.

- If the database instance is running and can be reached, select the DPA objects to remove

and then click Next.

Depending on the database type, you can remove one or both of the following objects:

- **Monitoring User:** You can remove the monitoring user if no other applications, including other installations of DPA, are using this user.
- **DPA Database Objects:** This refers to tables that are created in the schema of the monitoring user. If you remove the monitoring user, these objects are removed since they are owned by the monitoring user. You can remove these objects if no other installations of DPA are monitoring this instance.

 You cannot remove objects on certain database types, such as read-only replicas.

5. Confirm the unregistration information, and click Unregister Database Instance. This may take several minutes.
6. Click Finish to complete the unregistration.



# Database instance groups

See the following topics for information about creating and monitoring groups in DPA:

- [About monitoring SQL Server Availability Groups with DPA](#) describes the information that DPA provides about SQL Server AGs.
- [About monitoring Oracle multitenant databases \(CDBs\)](#) describes the information that DPA provides about Oracle CDBs.
- [Manually group database instances in DPA](#) describes how to create and modify custom groups.
- [View information about a group of database instances](#) explains how to view information about all instances in a group.

## About monitoring SQL Server Availability Groups with DPA

DPA provides status information, annotations, and alerts for your SQL Server Availability Groups (AGs).

- For information about options for registering SQL Server AGs, see [Registration and licensing options for clustered environments](#).
- DPA does not support monitoring distributed AGs. DPA can monitor the SQL Server instances that participate in a distributed AG, but the AG monitoring features are not enabled for distributed AGs.

## Automatic naming

When you [register an AG listener](#), DPA automatically names the instance using the following format:

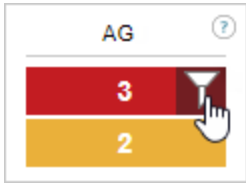
<PrimaryReplicaName> via <ListenerName>


When a failover occurs, the name is automatically updated to reflect the new primary replica.

- If you manually change the display name of an AG that is registered via the listener, by default DPA overwrites the name each time the monitor starts. To change the default behavior and manually specify the name, [change the advanced option](#) AG\_INSTANCE\_NAME\_UPDATE\_ENABLED.

## AG information in DPA

On the DPA homepage, the AG Status Summary box in the Status Summary area shows the number of database instances with partially healthy or not healthy AGs. As with other status boxes, click the filter next to a status to display only instances associated with the selected status.



The AG status icon  identifies database instances that include AGs. The color of the dot provides status information (described in the following section). Possible statuses are:

- Green for healthy
- Yellow for partially healthy
- Red for not healthy
- Gray for unknown

To view detailed information:

1. From the DPA homepage, click an AG status icon to open the Availability Group Summary view.  
This view shows information about each AG in the database instance. DPA shows status information for primary replicas in the instance. For secondary replicas, the status of the primary replica is displayed if DPA is also monitoring the primary replica.
2. Click any link to view detailed information about the databases and replicas in the AG.

## How DPA determines the AG status

If the instance is monitored directly and acting as a primary replica

If you are monitoring the instance directly (not through a listener), DPA looks at the status of all AGs that the instance acts as the primary replica for, and displays the worst status.

**Example:** An instance is acting as the primary replica for four availability groups. Their statuses are:

- AG1: Healthy
- AG2: Healthy
- AG3: Partially Healthy
- AG4: Not Healthy

DPA shows the status as **Not Healthy**.

AG1 (primary replica)	AG2 (primary replica)	AG3 (primary replica)	AG4 (primary replica)	DPA status
Healthy	Healthy	Partially Healthy	Not Healthy	Not Healthy

If the instance is monitored directly and acting as a secondary replica

If an instance is acting as a secondary replica for any AGs, that AG's status is Unknown. If the instance **also** acts as a primary replica for one or more AGs, the Unknown status is ignored.

**Example:** An instance acts as the primary replica for three availability groups. Their statuses are:

- AG1: Healthy
- AG2: Healthy
- AG3: Partially Healthy

The instance also acts as a secondary replica for one AG. Its status is Unknown.

DPA ignores the Unknown status, and shows the status as **Partially Healthy**.

AG1 (primary replica)	AG2 (primary replica)	AG3 (primary replica)	AG4 (secondary replica)	DPA status
Healthy	Healthy	Partially Healthy	Unknown	Partially Healthy

If DPA shows the AG status as Unknown, that typically indicates that the instance is acting as a secondary replica for all AGs.

If the instance is monitored via the listener

If you are monitoring the instance via the listener, by default DPA displays the aggregate status as described above. However, you can [change the advanced option](#) AG\_STATUS\_ROLLUP\_USE\_PRIMARY to determine the status using only the AG associated with the listener.

## AG alerts

DPA provides the following AG alerts: SQL Server Availability Group Failover and SQL Server Availability Group Status Change.

### SQL Server Availability Group Failover

This alert is triggered when an AG failover occurs. DPA sends alerts based on [how you registered instances](#):

- If you registered database instances directly (not through a listener), when a failover occurs DPA sends an alert for each instance involved in the failover that it is monitoring. For example,

if an AG fails over from Instance1 to Instance2 and DPA is monitoring both instances, you receive two alerts. If DPA is monitoring only one of the instances, you receive only one alert.

- If you registered the AG through a listener and the AG associated with the listener fails over, DPA sends one alert, because the listener moves with the AG from Instance1 to Instance2.

If multiple AG failovers occur in a short period of time, DPA aggregates them into one alert per instance.

## SQL Server Availability Group Status Change

This alert is triggered when an AG status changes to Partially Healthy or Not Healthy. DPA evaluates AG statuses every 10 minutes by default. You are alerted if the status changes from Healthy to Partially Healthy or Not Healthy between the evaluations. If the status changes from Healthy to another status and then back to Healthy during the same evaluation period, you are not alerted.

**Example:** In this example, DPA is monitoring an instance that acts as a primary replica for three AGs. The following table shows how the alerts would behave for each of DPA's alert policies. The intervals are arranged from most noisy to least noisy.

Interval	AG status (change in red)	Policy: Notify when level not visited since normal	Policy: Notify when level changes and is not normal	Policy: Notify when level is not normal
1	AG1 Healthy AG2 Healthy AG3 Healthy	No alert	No alert	No alert
8	<b>AG1 Healthy</b> AG2 Healthy AG3 Healthy	No alert	No alert	No alert
4	AG1 Healthy <b>AG2 Not Healthy</b> <b>AG3 Not Healthy</b>	AG2 Not Healthy AG3 Not Healthy	AG2 Not Healthy AG3 Not Healthy	AG2 Not Healthy AG3 Not Healthy

Interval	AG status (change in red)	Policy: Notify when level not visited since normal	Policy: Notify when level changes and is not normal	Policy: Notify when level is not normal
6	<p>AG1 <b>Healthy</b></p> <p>AG2 <b>Partially Healthy</b></p> <p>AG3 <b>Partially Healthy</b></p>	No alert	<p>AG2 Partially Healthy</p> <p>AG3 Partially Healthy</p>	<p>AG2 Partially Healthy</p> <p>AG3 Partially Healthy</p>
2	<p>AG1 Healthy</p> <p>AG2 <b>Partially Healthy</b></p> <p>AG3 Healthy</p>	AG2 Partially Healthy	AG2 Partially Healthy	AG2 Partially Healthy
3	<p>AG1 Healthy</p> <p>AG2 Partially Healthy</p> <p>AG3 <b>Partially Healthy</b></p>	AG3 Partially Healthy	AG3 Partially Healthy	<p>AG2 Partially Healthy</p> <p>AG3 Partially Not Healthy</p>
7	<p>AG1 <b>Partially Healthy</b></p> <p>AG2 <b>Healthy</b></p> <p>AG3 <b>Healthy</b></p>	AG1 Partially Healthy	AG1 Partially Healthy	AG1 Partially Healthy
5	<p>AG1 <b>Partially Healthy</b></p>	AG1 Partially Healthy	AG1 Partially Healthy	<p>AG1 Partially Healthy</p> <p>AG2 Not Healthy</p> <p>AG3 Not Healthy</p>

## Automatic annotations when AG failovers occur

Annotations are automatically added to wait time charts when an AG failover occurs. The annotations allow you to compare changes in performance before and after a failover.

The number of annotations depends on [how you registered instances](#):

- If you registered database instances directly (not through a listener), when a failover occurs DPA adds an annotation for each instance involved in the failover that it is monitoring. For example, if an AG fails over from Instance1 to Instance2 and DPA is monitoring both instances, DPA adds two annotations. If DPA is monitoring only one of the instances, DPA adds only one annotation.
- If you registered the AG through a listener and the AG associated with the listener fails over, DPA adds one annotation.

**i** If you do not want to add an annotation when an AG failover occurs, [change the value of the advanced system option](#) AG\_EVENT\_ANNOTATIONS\_ENABLED.

## About monitoring Oracle multitenant databases (CDBs)

If you are using DPA to monitor Oracle CDBs, see the following information about registration, grouping, and annotations.

### Registration and automatic grouping

To monitor an Oracle multitenant container database (CDB), register the pluggable databases (PDBs) contained in the CDB. Register each PDB just as you would register an Oracle single tenant database. For more information, see [Registration and licensing options for clustered environments](#).

When you register two or more Oracle PDBs in the same CDB, DPA automatically creates a group for the CDB. This group is used for all registered PDBs from the CDB. If a DBA moves a PDB to a new CDB, DPA processes and groups the instance.

### View the PDB load

On the DPA homepage, you can:

- Click the CDB name to [view summary data](#) from all PDB instances in the group. Use this view to determine which PDB has the most wait time and what types of waits the PDBs are experiencing.
- Expand the CDB group and click a PDB name to view activity and [investigate performance issues](#) on that database instance.

## Automatic annotations

Annotations are automatically added to wait time charts when a PDB is added, removed, or moved from one CDB to another. The annotations allow you to compare performance before and after the change.

## Turn off automatic grouping of Oracle CDBs

If you do not want DPA to automatically group the PDBs within a CDB, you can turn automatic grouping off.

1. Click Options.
2. Under Administration > Configuration, click Advanced Options.
3. Click the ORACLE\_CDB\_AUTO\_GROUP system option.
4. Select False from the New Value list, and click Update.

After you set this option to false, grouping of registered database instances does **not** change. Only newly registered or updated database instances are affected, and are not grouped.

## Manually group database instances in DPA

DPA automatically groups Oracle Real Application Clusters (RAC) instances and Oracle multitenant container databases (CDB) containing pluggable databases (PDB). You can manually group other database instances so that they are displayed together on the DPA homepage. For example, you can create groups based on type or location. When database instances are grouped, you can [view information](#) about all instances in the group.

A database instance can be included in only one group.

## Create a custom group

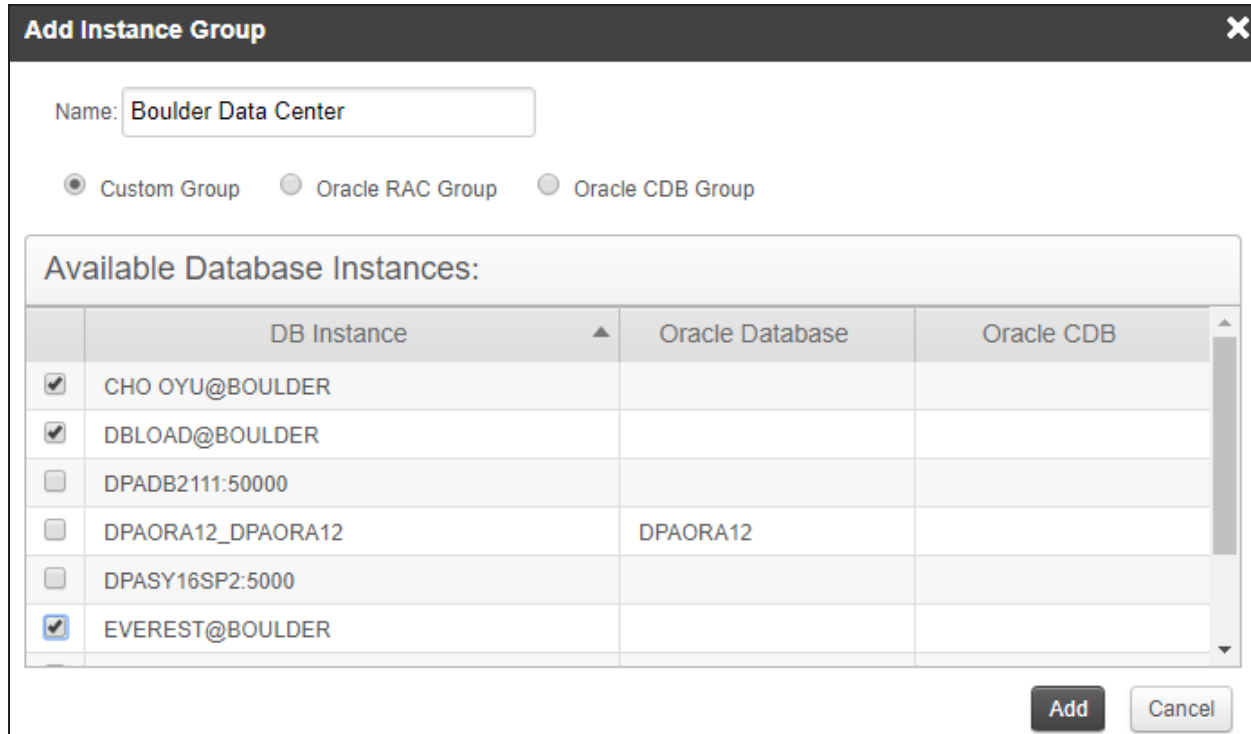
1. On the DPA homepage above the list of database instances, click Group Settings.

The Manage Instance Groups dialog box lists the existing groups.

2. Click Add.

The Add Instance Group dialog box lists the database instances that are not members of an existing group.

3. Enter a name, select the database instances to include, and then click Add.



**Add Instance Group** ✕

Name:

Custom Group
  Oracle RAC Group
  Oracle CDB Group

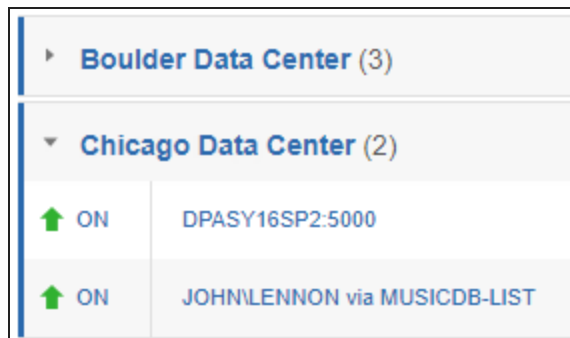
Available Database Instances:

	DB Instance ▲	Oracle Database	Oracle CDB ▲
<input checked="" type="checkbox"/>	CHO OYU@BOULDER		
<input checked="" type="checkbox"/>	DBLOAD@BOULDER		
<input type="checkbox"/>	DPADB2111:50000		
<input type="checkbox"/>	DPAORA12_DPAORA12	DPAORA12	
<input type="checkbox"/>	DPASY16SP2:5000		
<input checked="" type="checkbox"/>	EVEREST@BOULDER		

## Show or hide groups on the DPA homepage

Toggle the Show Groups button above the list of database instances to show or hide groups.

- When you show groups (the default), the DPA homepage lists ungrouped database instances first, followed by groups in alphabetical order. You can expand or collapse each group. Click the group name to [view information](#) about all instances in the group.



▶	<b>Boulder Data Center (3)</b>
▼	<b>Chicago Data Center (2)</b>
↑ ON	DPASY16SP2:5000
↑ ON	JOHNLENNON via MUSICDB-LIST

- When you hide groups, the DPA homepage lists database instances alphabetically.

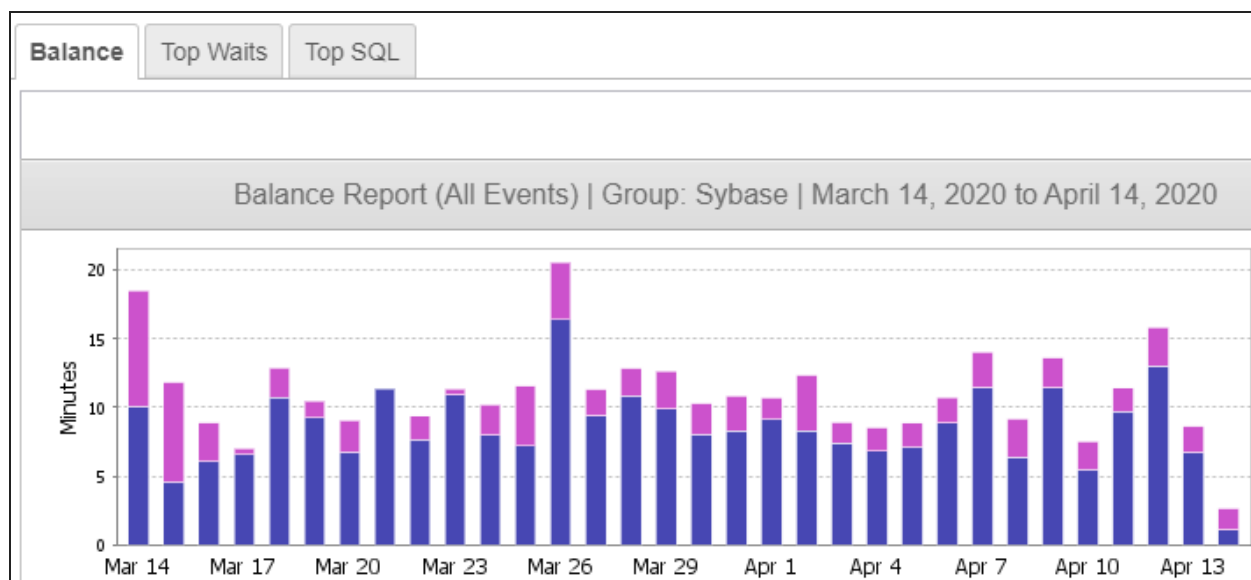


## View information about a group of database instances

DPA automatically groups Oracle Real Application Clusters (RAC) instances and Oracle multitenant container databases (CDB) containing pluggable databases (PDB). You can also [manually create groups](#) of database instances. When databases are grouped, you can view information about how wait time is distributed throughout the group and top waits and top SQL statements for the entire group.

1. From the DPA homepage, click the name of the group.

The Balance Report bar graph shows the amount of wait time for each database instance in the group for the past month. Use this report to evaluate load distribution among the group members.




2. Click Top Waits to see the top 15 waits across all database instances in the group.
3. Click Top SQL to see the 15 SQL statements with the most wait time across all database instances in the group.

## Monitor database instances with DPA

After you [register database instances for monitoring](#), the DPA homepage displays a list of the monitored database instances with status and wait time information. Click Action to start or stop monitoring.

Monitoring is always active after it is started. It is not necessary to restart the DPA monitor if the repository instance or the monitored database instance was unavailable for a period of time. Monitoring resumes when both are available again.

- If there is a period of time when monitoring should not occur, you can [stop and then restart monitoring](#).
- If connection or user information changes with one of your monitored database instances, you can [update that information](#).
- If you are having problems connecting to or monitoring a database instance, see [Troubleshooting tips](#).

 For more information about using DPA to resolve issues on monitored instances, see [Investigate performance issues with DPA](#).

## Update a monitored database instance

If connection or user information changes with one of your monitored database instances, you must update that information in DPA.

1. On the DPA menu, click Options.
2. Under Monitor Setup > Database Instances, click Update Connection.
3. Select the database instance, and click Next.
4. Select the check box next to the property, update the value, and click Next.

For database-specific connection information, see the following:

- [Oracle](#)
- [SQL Server](#)
- [Azure SQL](#)
- [Sybase](#)
- [Db2](#)
- [MySQL](#)
- [Amazon RDS for Oracle](#)
- [Amazon RDS for SQL Server](#)
- [Amazon RDS for MySQL](#)

5. Confirm the connection information, and click Update Connection.
6. Click Finish, or Update Another Database Instance to continue updating.

## Stop monitoring a database instance for a period of time

A blackout is a period of time when DPA stops monitoring a certain database instance.

1. On the DPA menu, click Options.
2. Under Monitor Setup > Database Instances, click Monitor Blackout Periods.
3. Select a database instance from the list on top.
4. Set a day and time to stop and start monitoring, and click Add New Blackout Period.

## DPA troubleshooting tips

### Logs

DPA logs information about each monitored database instance. Use this information to help you determine why a database instance is not being monitored, or if data are missing.

#### Access log data through the DPA log viewer

Use the DPA Log Viewer to view log information for a specific database instance, or for all database instances and the DPA repository.

1. Open the Log Viewer:
  - To display log messages for a specific database instance:  
From the DPA homepage, click Action > Log next to the database instance.
  - To display messages for all monitored database instances and the DPA repository:  
On the DPA menu, click Options. Then, under Support > Utilities, click Log Viewer.
2. Use any of the following options to locate information:
  - Use filters to help you find specific information. To change the filters, click Advanced and select the filter criteria. For example, you can filter by date range, a text string, or message level.
  - For any message above Info, click Details to view additional information from the log.
  - Click Log Files for Support to create a compressed file you can send to SolarWinds Support.

#### Open log files in a text editor

Log files are stored in the `installDir/iwc/tomcat/logs/` directory.

### Access to a database instance

If DPA cannot access the server that hosts a database instance you want to monitor:

- Make sure a firewall is not running on the server.
- Make sure another process is not using the default DPA [ports](#).

If the ports are being used by another process, you can change the default ports of 8123, 8124, and 8127. To specify different ports for DPA to use:

1. Open the following file in a text editor:

```
installDir/iwc/tomcat/conf/server.xml
```

2. Update the following lines with new port numbers:

```
<Server port="8127" shutdown="SHUTDOWN">  
  <Connector port="8123"/>  
  <Connector port="8124"/>
```

3. Save the file and restart DPA.

## Issues after the Oracle PDB that stores the repository is moved

If the DPA repository is created on an Oracle pluggable database (PDB), you might experience the following issues after the PDB is moved to a different container database (CDB).

### DPA returns a connection error

The PDB moved to a CDB on a different server, and the connection string is incorrect. Update the connection string in the `repo.properties` file in the following location:

```
<DPA_Install_Dir>\iwc\tomcat\ignite_config\iwc\repo.properties
```


### DPA returns an invalid login error

Verify that the DPA monitoring user exists in the CDB. Common users (prefaced with C##) exist in only one CDB.

# Investigate performance issues with DPA

DPA uses an approach called [wait-based analysis](#) to help you focus on issues that provide the greatest performance improvements. Query advisors and table tuning advisors identify performance issues and help you find the root cause:

- [Access DPA query or table tuning advisors](#)
- [Use DPA's query performance analysis to find the root cause of performance issues](#)
- [Investigate inefficient queries running against a table](#)
- [Identify blocking sessions and deadlocks with DPA](#)
- [Find and investigate unusually long wait times \(anomalies\)](#)
- [About anomaly detection in DPA](#)
- [Add an annotation to document a change to the database](#)

 The DPA Getting Started Guide includes walk-through examples of using DPA to investigate performance problems:













- [Investigate an application performance problem](#)
- [Investigate an increase in wait time](#)

## Access DPA query or table tuning advisors

DPA provides two types of advisors:

- **Query advisors** provide information to help you improve the performance of a specific query, including what type of waits were responsible for significant wait time, whether the statement was blocked by other sessions, and whether execution plans include potentially expensive steps such as full table scans.
- **Table tuning advisors** are generated when a significant number of inefficient queries run against a table. These advisors provide aggregated information about the table, the inefficient queries that ran against it, and any existing indexes.

The Tuning column on the DPA homepage displays a warning or critical icon when advisors with a warning or critical status are available for a database instance. A green check mark in this column indicates that there are no advisors or that all advisors are informational.

Database Instance ▲		Wait	Tuning	CPU	Mem	Disk	Sess
AVANTIA@BOULDER	Action ▼						
AVANTIO	Action ▼						

## View all advisors for a database instance

To view all advisors for a database instance, do either of the following to open the Tuning Advisors page:

- From the DPA homepage, click the icon in the Tuning column.
- If you have drilled in to view information about a database instance, click the Tuning tab in the top-right corner of the instance details page.



A red or yellow bar on the Tuning tab indicates that critical or warning advisors are available.

The Tuning Advisors page displays the latest query and table tuning advisors. Use the drop-down menu at the top of the page to display advisors generated for a previous date.

**i** Query advisors are calculated every hour. Table tuning advisors are calculated once a day, at the end of the day. The most recent table tuning advisors are for the previous day.

## Open an advisor

For detailed information to help you resolve performance issues:

- Click a query advisor to open the [Query Detail page](#), which displays detailed information about the query along with the most relevant statistics and metrics charts.
- Click a table tuning advisor to open the [Table Tuning Advisor page](#), which displays aggregated information about the table and the inefficient queries that ran against it.

## Use DPA's query performance analysis to find the root cause of performance issues

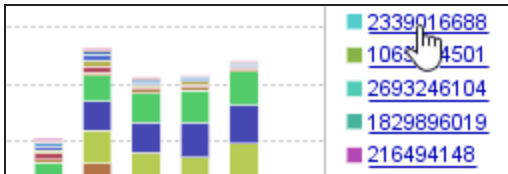
To help you investigate the root cause of a query's performance problems, DPA intelligently assembles the most relevant data about the query and displays it on the Query Details page. Use the Query Details page to:

- View waits, statistics, and metrics from [any time period](#)
- See [what type of waits](#) are affecting performance
- Review [query and table tuning advisors](#)
- Examine [statistics and metrics charts](#) to correlate query wait times with other events

**i** See an example of using the Query Details page to [investigate an increase in wait time](#).

## Open the Query Details page

Click the SQL hash or name in any chart legend to open the Query Details page.



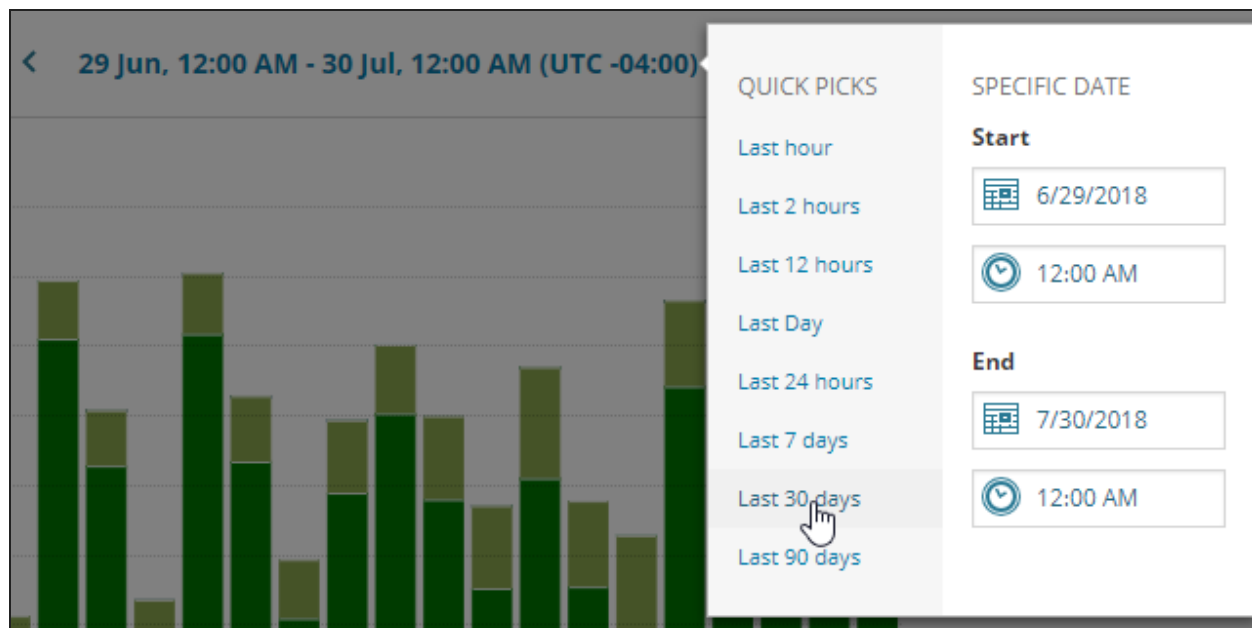
## Select a time period

All data on the Query Details page reflects the selected time period, which is displayed at the top of the page.

When you open the Query Details page, it defaults to the time period selected for the previous chart. For example, if you open the Query Details page while viewing the Top SQL Statements for one day, the Query Details page shows data for that day.

To select a different time period, you can:

- Click a bar to drill in to that time period.
- Click the date range at the top of the page to open the date picker. Then select a predefined time period or enter specific dates.




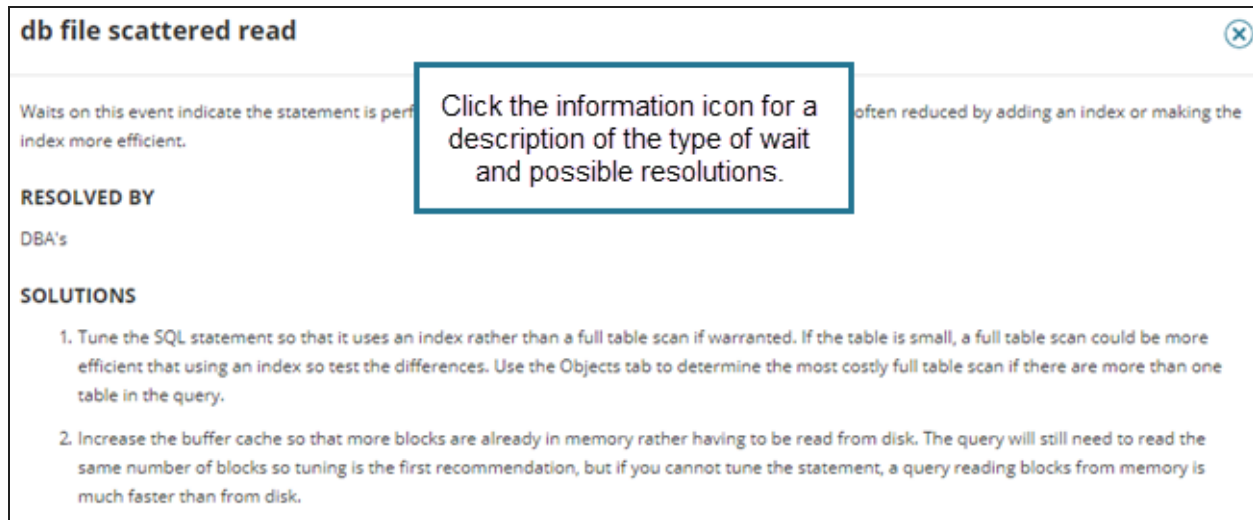


## See what type of waits are affecting performance

The Top Waits chart at the top of the page shows the query's execution time for the selected time period. The bars are color-coded by the type of wait. Knowing what type of waits are causing the performance issue can help you determine how to fix the issue.

On this chart, you can:

- Click the  next to an entry in the legend to display detailed information about that type of wait, including possible resolutions.



**db file scattered read** ✕

Waits on this event indicate the statement is performing a full table scan. This is often reduced by adding an index or making the index more efficient.

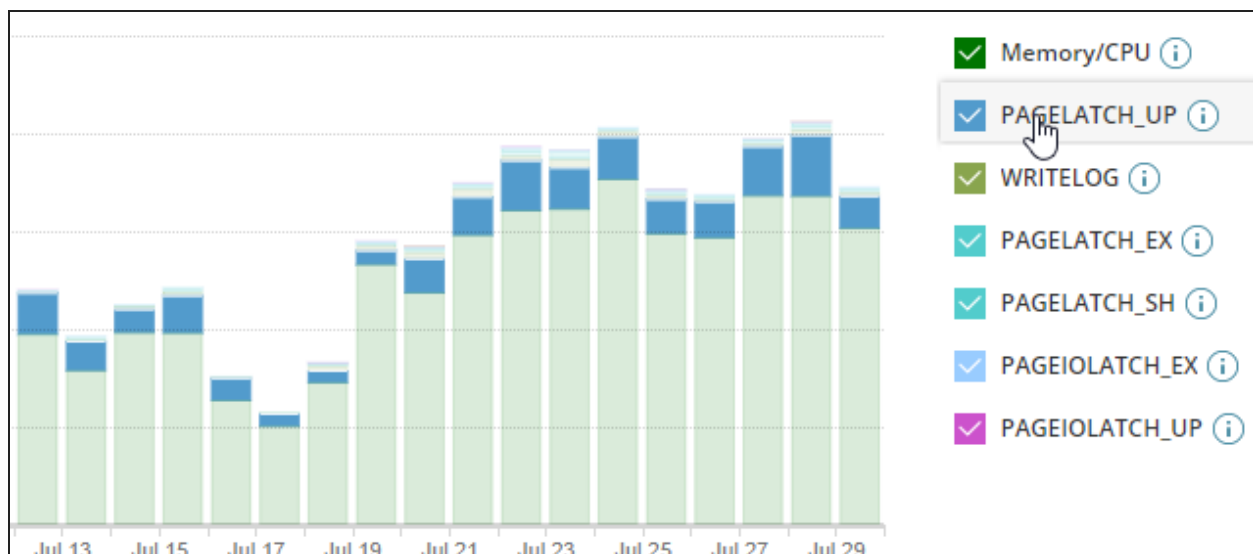
Click the information icon for a description of the type of wait and possible resolutions.

**RESOLVED BY**  
DBA's

**SOLUTIONS**

1. Tune the SQL statement so that it uses an index rather than a full table scan if warranted. If the table is small, a full table scan could be more efficient than using an index so test the differences. Use the Objects tab to determine the most costly full table scan if there are more than one table in the query.
2. Increase the buffer cache so that more blocks are already in memory rather than having to be read from disk. The query will still need to read the same number of blocks so tuning is the first recommendation, but if you cannot tune the statement, a query reading blocks from memory is much faster than from disk.

- Hover over an entry in the legend to dim other waits in the chart and better visualize the impact of this type of wait.



## Review query and table tuning advisors

The Query Advisors section shows the latest advice for the selected time period. Query advisors provide information such as:

- What type of wait activities the SQL statement spend significant time on.
- Whether the statement was blocked by other sessions.
- Whether the statement took longer than normal to execute.
- If multiple execution plans were used, or if plans include potentially expensive steps such as full table scans.

If any [table tuning advisors](#) included information about this query, you can click through for aggregated information about the table and all inefficient queries that ran on it.

## Correlate query wait times with other events

To help you find the root cause of performance issues, the Query Details page includes the most relevant statistics, blocking, plan, and metrics charts. Sections with data to display are automatically expanded. Other sections are collapsed by default. For example, if there is no blocking data, the Blocking section is collapsed.

When you scroll down to view these charts, the Top Waits chart at the top of the page remains visible so you can correlate query wait times with other events during the same time period.



DPA uses the predominant type of wait and other information to automatically select the most relevant charts. For example, if the predominant type for an Oracle database instance is Memory/CPU, DPA includes charts such as OS/CPU Utilization, CPU Utilization by DB, and Buffer Cache Hit Ratio.

**i** The *predominant type of wait* is the type responsible for the majority of the time that a query spent waiting during the specified period.

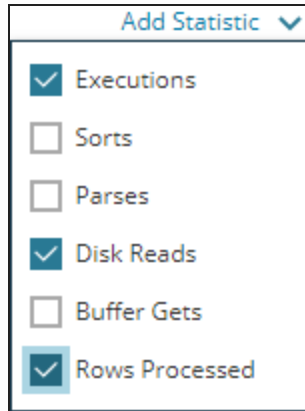
To be considered predominant, the type must be responsible for more than a certain percent of the total wait time for that period. By default, this threshold is 20%. You can change the threshold by [changing the advanced option](#) PREDOMINANT\_WAIT\_THRESHOLD.

You can manually select other statistics or metrics charts to include.

### Display other statistics charts

1. If the Statistics section is collapsed, expand it.
2. On the right side of the Statistics section, click Add Statistic.

3. Select the statistics you want to include, and deselect any you want to remove.



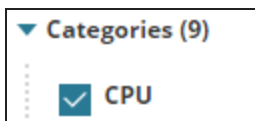
4. Click outside the drop-down to close it.

## Display other metrics charts

1. If the Instance Resource Metrics section is collapsed, expand it.
2. On the right side of the section, click Add Metrics.

The Add Metrics dialog box opens.

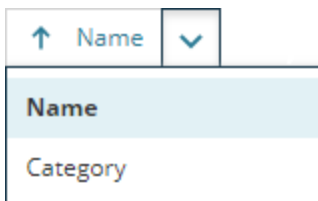
3. Filter or sort the list to locate the metrics you want to add:
  - Select one or more categories to filter by those categories.



- Enter a string in the Search box to show only metric names containing that string. (Wildcards are not supported.)

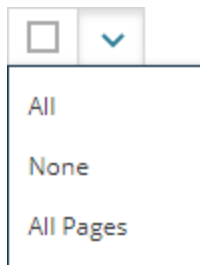


- Sort by name or by category.



4. Select one or more metrics. Use the selection drop-down menu to quickly select multiple metrics:

- All: Selects all metrics on the current page.
- All Pages: Select all metrics on all pages.



5. Click Save Changes to display the selected metrics.

## Investigate inefficient queries running against a table

Inefficient queries—that is, queries that perform a large number of reads but return a relatively small number of rows—can significantly add to database performance issues. These queries do a large amount of work for little return. This type of inefficiency results in higher I/O, longer wait times, greater amounts of blocking, and increased resource contention.

Possible solutions include tuning the query, adding an index, or adding columns to an existing index. DPA's **table tuning advisors** help you make informed decisions about the best course of action.

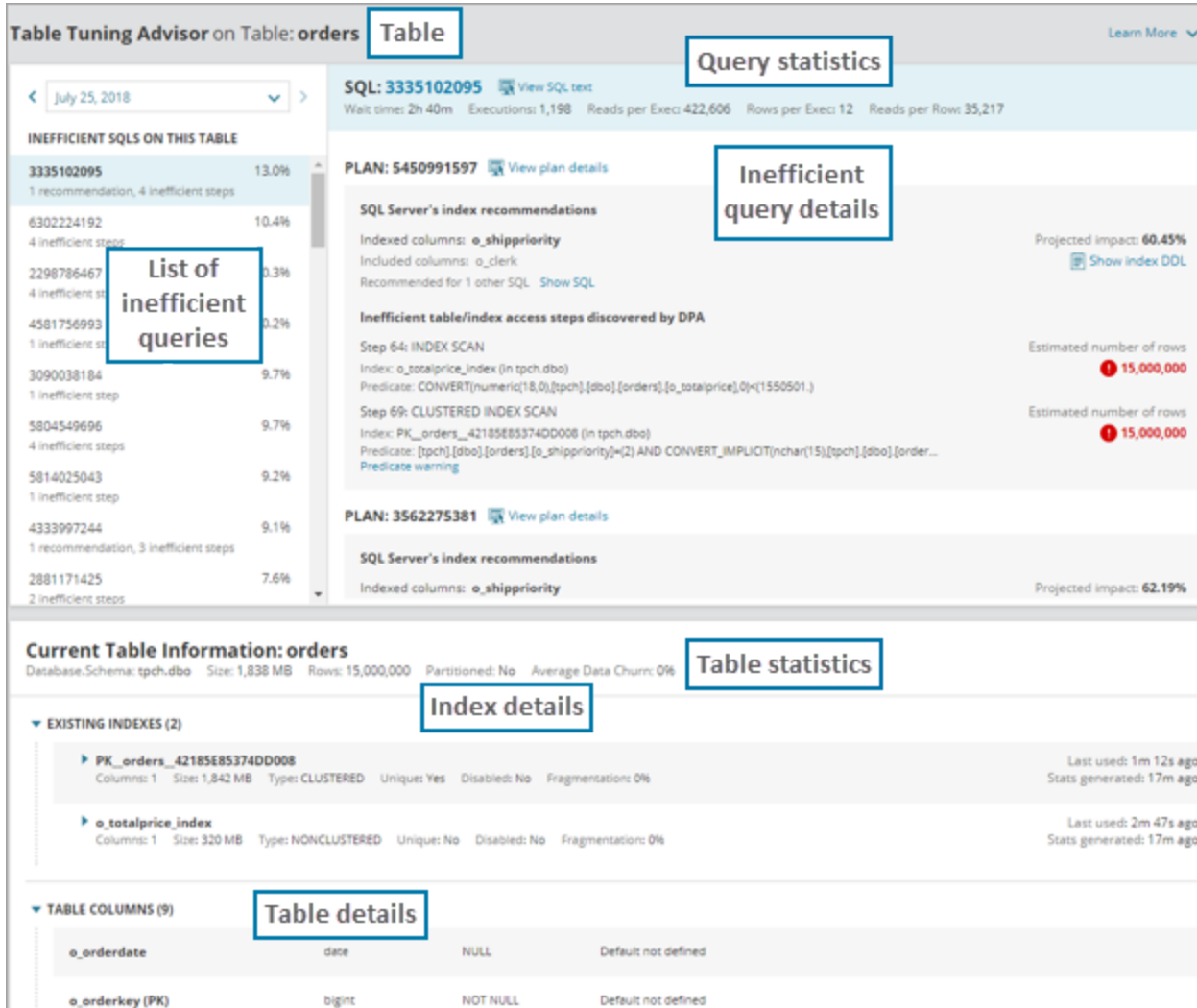
See the following sections for tips on using the information in each table tuning advisor:

- [What are table tuning advisors?](#)
- [Open a table tuning advisor](#)
- [Quick start](#)
- [Examine the list of inefficient queries](#)
- [Examine query details](#)
- [Examine table statistics](#)
- [Examine index details](#)

### What are table tuning advisors?

At the end of each day, DPA runs an analysis to identify tables that had inefficient queries run against them during that day. For each of these tables, the Table Tuning Advisor page displays aggregated information about the table, the inefficient queries that ran against it, and any existing indexes. This information helps you optimize query performance while taking indexing trade-offs into account.

- Table tuning advisors are available for Oracle, SQL Server (2008 and above), and Azure SQL databases.
- Table tuning advisors are calculated at the end of each day. Therefore, the most recent table tuning advisors are for the previous day.



The screenshot displays the 'Table Tuning Advisor on Table: orders' page. It is divided into several sections:

- Table:** A dropdown menu showing 'orders'.
- Query statistics:** Shows SQL: 3335102095, Wait times: 2h 40m, Executions: 1,198, Reads per Exec: 422,606, Rows per Exec: 12, Reads per Row: 35,217.
- INEFFICIENT SQLS ON THIS TABLE:** A list of queries with their respective inefficiency percentages. A callout box labeled 'List of inefficient queries' points to this list.
- SQL Server's index recommendations:** Shows indexed columns (o\_shippriority) and included columns (o\_clerk). A callout box labeled 'Inefficient query details' points to this section.
- Inefficient table/index access steps discovered by DPA:** Details steps like 'INDEX SCAN' and 'CLUSTERED INDEX SCAN' with estimated row counts (15,000,000).
- Current Table Information: orders:** Shows database schema (tpch.dbo), size (1,838 MB), rows (15,000,000), and other statistics. A callout box labeled 'Table statistics' points to this section.
- EXISTING INDEXES (2):** Lists indexes like 'PK\_orders\_42185E85374DD008' and 'o\_totalprice\_index' with their properties. A callout box labeled 'Index details' points to this section.
- TABLE COLUMNS (9):** Lists columns like 'o\_orderdate' and 'o\_orderkey (PK)' with their data types and constraints. A callout box labeled 'Table details' points to this section.

## Open a table tuning advisor

The [Tuning tab](#) lists all table tuning advisors for the selected database instance. Click a table tuning advisor to open it.

## Quick start

Each table tuning advisor provides detailed information, as described in the following sections. Use the following suggestions to get started:

## 1. Update statistics.

In the Existing Indexes section, look at the age of the index statistics. If the statistics are stale, especially if table churn is high, the optimizer does not have the best information to make good plan choices. Updating statistics is often a good first step before you do any further analysis.

## 2. Evaluate indexes.

Click on several of the top inefficient queries and do the following:

- Review the SQL text to learn more about the WHERE clauses and JOIN conditions that can affect query performance.
- (SQL Server and Azure only) If plans with SQL Server's index recommendations are provided, consider adding them or extending existing indexes to satisfy them.
- If plans with inefficient table or index access steps are provided:
  - Review each plan section and the predicates for each step. The columns in the predicates are candidates for indexes.
  - Check for warnings (shown as links below the step if they are detected) and consider their recommendations.
  - Consider indexing the candidate columns found across the SQL statements examined:
    - Is there an index that might benefit several queries?
    - Is there an existing index that could be extended to benefit one or more queries?

## 3. Resolve fragmentation.

Review the table's row count, churn, and index fragmentation. For larger tables, consider the following:

- If fragmentation is high, defragmenting the indexes might help resolve performance problems when plan steps are using scan operations.
- If churn is also high, consider defragmenting the index more frequently.

## Examine the list of inefficient queries

The top-left pane lists the inefficient queries that ran against the table on the selected day. DPA assigns a relative efficiency score to each query and uses this score sort the list.

Select a query from this list to display detailed information about it.

< July 23, 2018 >

**INEFFICIENT SQLS ON THIS TABLE**

<b>3335102095</b>	15.6%	← Least efficient query
1 recommendation, 4 inefficient steps		
4333997244	11.0%	
1 recommendation, 3 inefficient steps		
5814025043	9.2%	
1 inefficient step		
2881171425	9.2%	
2 inefficient steps		
3090038184	9.1%	
1 inefficient step		


## Tips for using this information

- Focus your tuning efforts on the queries at the top of the list, which are driving the most inefficient workload against this table.
- A large number of queries in the list could indicate a more widespread performance issue. Perhaps one good index could improve the performance of several similar queries.


## Examine query details

The upper-right pane displays information about the selected query that can help you determine the source of read inefficiencies against this table.




**SQL: 3335102095**  View SQL text 1


Wait time: 3h 35m Executions: 1,638 Reads per Exec: 424,807 Rows per Exec: 12 Reads per Row: 35,401

**PLAN: 5450991597**  View plan details 2

**SQL Server's index recommendations** 3

Indexed columns: **o\_shippriority** Projected impact: **60.45%**  
Included columns: o\_clerk  Show index DDL  
Recommended for 1 other SQL [Show SQL](#)

**Inefficient table/index access steps discovered by DPA** 4

Step 64: INDEX SCAN  Estimated number of rows  
Index: o\_totalprice\_index (in tpch.dbo) **15,000,000**  
Predicate: CONVERT(numeric(18,0),[tpch].[dbo].[orders].[o\_totalprice],0)<(1550501.)

Step 69: CLUSTERED INDEX SCAN Estimated number of rows  
Index: PK\_orders\_42185E85374DD008 (in tpch.dbo) **15,000,000**  
Predicate: [tpch].[dbo].[orders].[o\_shippriority]=(2) AND CONVERT\_IMPLICIT(nchar(15),[tpch].[dbo].[order...]  
[Predicate warning](#)

- 1 The performance statistics at the top of the pane show the extent of the query's inefficiency:
  - Reads per Exec is the number of read I/O operations per execution, which indicates how much work the query is doing.
  - Rows per Exec is the number of rows the query returns.
  - The Reads per Row ratio is the number of reads the query needed to do in order to arrive at each row in the query's result set. Statements with the highest Reads per Row ratios could potentially benefit most from tuning.

For more information about the query, click the SQL name or hash value to view DPA's [query performance analysis](#), which shows when the query ran, the execution statistics, and the most relevant metrics charts.

- 2 DPA lists each execution plan that it finds. You can click the link to examine the full plan, but DPA lists the steps most likely to need attention below.
- 3 (SQL Server and Azure only) Index recommendations made by the SQL Server optimizer, if any, are listed. The Projected Impact is the cost reduction that the optimizer estimates the recommended index will have. Click Show index DDL to see the CREATE INDEX statement for the recommendation.

#### 4 DPA analyzes the plan and lists steps with the most inefficient access paths.

**i** These steps read data to be processed by subsequent "consumer" plan steps. While consumer steps (for example, sorts) can have a high plan cost, they are usually affected by a preceding step that read too much data.

Information about each step includes:

- The step number and the type of operation being performed in the step (for example, INDEX SCAN).
- The index this step uses, if it uses an index.
- Any predicates. These are snippets of the SQL that the plan step is acting on. They are typically portions of JOIN or WHERE clauses in which a table's column is being compared to another column or value.
- Any warnings that apply to the step.
- The number of rows the optimizer estimates this step will read. Critical and warning icons identify steps that read a high percentage of the table or index rows, and therefore have a greater need for tuning or an index.

#### Tips for using this information

- Before you add an index, weigh the projected impact or potential performance improvement against [indexing trade-offs](#). Also consider the indexing needs of other queries.
- Click any step to get detailed information about the operation and recommendations for potentially reducing the amount of I/O.



- If predicates are listed, they often indicate which columns need to be indexed, or where the optimizer is not using an existing index. For example, if the query calls a function on the column, the plan will not use an index.
- If warnings are listed, click the warning for a detailed description of the condition that DPA has identified as a potential reason for concern.

#### Warnings

DPA provides the following warnings:

- A **predicate warning** occurs when a column needs to be converted to a different data type before it can be used. For example, if a query has a JOIN clause that equates a numeric column to a varchar column, one of the columns will be implicitly converted to the other's type. The

optimizer typically does not use an index on an implicitly converted column. This is often why the optimizer doesn't use an existing index that the query's author expected it to use.

- A **lookup warning** typically indicates that the database is doing an index lookup to identify the target rows, then doing an extra table access to get data not found in the indexed columns. To get better performance, consider adding a covering index, or extending an existing index to include columns needed to avoid the table lookup. However, remember that adding a large number of columns can increase the [index size and maintenance overhead](#).
- A **spool warning** indicates that the step's result set is being stored for reuse later in the query's execution. While spool operations are often beneficial, the intermediate data storage can cause disk overhead and contention.
- A **parallel warning** indicates that DPA has detected a parallelism step later in this query's execution, implying that this step's intermediate result set is likely large enough to exceed parallel processing cost thresholds. Look for ways to rewrite the query to reduce the size of intermediate result sets earlier in the query. For example, look for a sub-select that could produce fewer rows or the nested loop join order if more than two tables are involved.

## Examine table statistics

At the top of the Current table information section, DPA provides table statistics, such as the size of the table and the amount of churn.

<b>Current Table Information: CON_ALERT_HISTORY</b>		<b>1</b>	<b>1 of 5 table tuning best practices not fulfilled</b>
Schema: HUFFYO	Size: 28 MB	Rows: 218,760	Used Blocks: 3,520
Block Size: 8 KB	Stats Generated: 768d ago	Partitioned: No	Average Data Churn: 15%
<b>2</b>	<b>3</b>		<b>4</b>

### Tips for using this information

- 1 Best practices:** If the table or its indexes do not fulfill all of DPA's best practice recommendations, click the info icon to find out which recommendations are not met. [Click here](#) for information about correcting any violations.
- 2 Size and Rows:** For large tables, indexing is often critical to good query performance, although an index on a large table uses large amounts of disk space. For small tables, full table scans sometimes offer better performance than the use of indexes.
- 3 Stats Generated:** If the statistics are old and data churn is high, statistics should be updated frequently to provide the optimizer with the information it needs to make better plan decisions.

- 4 Churn:** A table's churn is the daily number of insert and delete operation expressed as a percentage of the total number of table rows.

Each insert and delete statement, as well as some update statements, incur a performance hit due to index maintenance. Generally, the higher the churn, the more caution you should take when adding an index. Before you add a new index, weigh the query execution time saved against the time spent on index maintenance.

## Examine index details

DPA displays information about all existing indexes on the table, including the structure, the amount of fragmentation, how long ago the statistics were generated, and when the index was last used.

### Where does DPA get the last used value for an index?

For SQL Server database instances, DPA shows when the index was last used for a seek, scan, or lookup operation, which is recorded in the `sys.dm_db_index_usage_stats` table. This value is not updated as a result of system activity.

For Oracle databases, DPA shows when the index was last included in an Oracle execution plan for a select, update, insert, or delete statement.

### Tips for using this information

Before you make any indexing decisions, first review the existing indexes. Consider the following questions.

**i** Take [indexing trade-offs](#) into account when you are considering adding or extending an index.

- Are the statistics stale? If the statistics are old and data churn is high, statistics should be updated frequently to provide the optimizer with the information it needs to make better plan decisions.

If statistics are old and churn is high, consider updating the statistics before adding or modifying indexes.

- Is there an existing index that an inefficient query should be using?

Look for ways to adjust the query so that it uses the index.

- If an inefficient query is using an existing index, are there inefficient table or index access steps on columns that aren't included in the index?

Consider adding those columns to the existing index.

- If an inefficient query is using an existing index, are there inefficient table or index access steps

that indicate a [lookup warning](#)?

Consider adding those columns to the existing index to make it a covering index for the query.

- Is there no existing index that would improve an inefficient query's performance?

Consider adding a new index.

- Are indexes fragmented? Fragmentation occurs as a result of numerous insert and delete statements. Fragmentation causes index data to become out of order on the disk, with gaps between index data. This is not a major concern for small tables, but for large tables this can cause slow performance when the index is read using a scan operation.

Consider defragmenting your indexes on a regular basis for large tables, especially if data churn is high and many scans are occurring.

## Correcting common index problems

After you determine what indexes are needed to improve query performance, look for additional benefits by identifying poor index usage, such as:

- **Unused indexes:** Can indexes be removed without negatively affecting query performance? To help you find unused indexes, DPA lists how long ago each index was used. However, before you remove an index:
  - Be aware that sometimes the Last Used value can show only the date since the monitored database instance was last started.
  - Consider whether queries that run infrequently (for example, monthly or quarterly) might use the index.
- **Too many indexes:** A large number of indexes on a table might be necessary for important queries to run quickly. However, you should also consider the performance overhead of index maintenance on other DML statements. Look for opportunities to:
  - Combine similar indexes.
  - Remove unused or rarely used indexes.
  - Remove indexes that were added for queries that are not performance sensitive.
- **Overlapping indexes:** Two indexes overlap if they both have the same leading edge columns in the exact same order, but one index has at least one additional column at the end. In this case, the larger index (with more columns) is all that you need, and you can remove the smaller, redundant index. Alternatively, you might choose to remove the larger one if the additional columns are not being used, or if the additional columns offer little benefit compared to the cost of index maintenance.
- **Questionable index structure:** The following might indicate a poorly-constructed index:

- Many columns: Indexes with many columns require more storage, and increase the cost of index maintenance. Perhaps the index was defined this way to make it a "covering index" for some queries. If not, consider removing trailing edge columns.
- Wide columns: Some DBAs question the benefit of adding wide columns (for example, long varchars) to an index, because of the high amounts of storage needed for the index and the maintenance overhead. With this in mind, if your queries do a lot of searching on any column, consider indexing it.

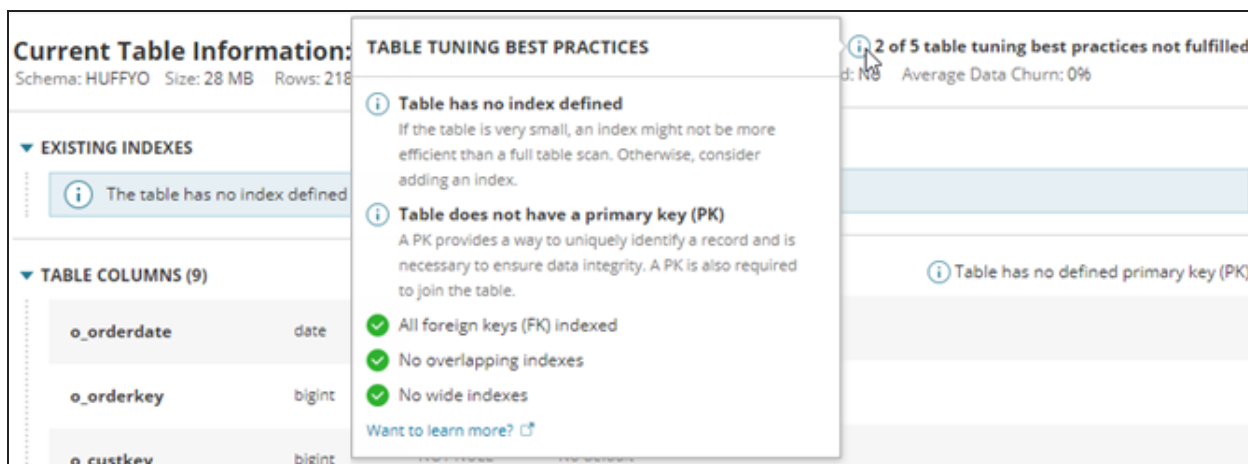
## Indexing trade-offs

While indexes can provide performance benefits for some queries, consider the following trade-offs when making indexing decisions:

- **Index maintenance:** When a table row is inserted or deleted, the corresponding entry in each index must also be inserted or deleted. If an indexed column is updated, the associated entries in the index must also be updated. These operations on indexes increase the time an insert, delete, or update statement takes to run. The cost of index maintenance increases as the amount of data churn increases.
- **Disk space:** Indexes consume disk space. The larger the table and the more columns in the index, the more disk space it needs.

## Table tuning best practices

When DPA generates a [table tuning advisor](#), it evaluates the table and its indexes against a set of best practices. Any violations are listed in the Current Table Information section.



**Current Table Information:**  
Schema: HUFFYO Size: 28 MB Rows: 218

**EXISTING INDEXES**

The table has no index defined

**TABLE COLUMNS (9)**

o_orderdate	date
o_orderkey	bigint
o_custkey	bigint

**TABLE TUNING BEST PRACTICES**

2 of 5 table tuning best practices not fulfilled  
Average Data Churn: 0%

- ❗ **Table has no index defined**  
If the table is very small, an index might not be more efficient than a full table scan. Otherwise, consider adding an index.
- ❗ **Table does not have a primary key (PK)**  
A PK provides a way to uniquely identify a record and is necessary to ensure data integrity. A PK is also required to join the table.
- ✅ All foreign keys (FK) indexed
- ✅ No overlapping indexes
- ✅ No wide indexes

Want to learn more? [🔗](#)

If violations are found, consider the following recommendations.

**i** Use the following [advanced configuration options](#) to change the default values that DPA uses to check for best practices:

- To prevent DPA from checking for compliance to a best practice, change the corresponding `BEST_PRACTICES_<practiceName>` option to `false`.
- Use `BEST_PRACTICES_WIDE_INDEX_SIZE` to change the minimum size of a wide index.
- Use `BEST_PRACTICES_NUMBER_OF_COLUMNS_IN_WIDE_INDEX` to change the minimum number of columns in a wide index.
- Use `BEST_PRACTICES_NUMBER_OF_OVERLAPPING_COLUMNS` to change the minimum number of leading edge columns that indexes must share to be classified as overlapping.

## Foreign key (FK) is not indexed

A FK in one table (the child table) refers to the primary key of another table (the parent table). Indexing each FK can improve the performance of queries that join the two tables. In addition, when a FK is not indexed, the database must perform a full table scan of the child table whenever a row is deleted or the primary key value is updated in the parent table.

## Overlapping indexes found: At least two indexes have the same leading edge columns

Overlapping indexes have the same leading edge column (the first column defined). Because every index has a [maintenance cost](#) and consumes disk space, identifying and removing unneeded indexes can improve performance. Examine the overlapping indexes to determine if any can be removed. For example:

- If two indexes include the same columns in the same order but one includes additional columns, the smaller index is redundant and can be removed.
- If two indexes include the same columns but each has one additional column, modify one index to include all columns and remove the other index.

## Wide index found: Index contains five or more columns or is more than 200 bytes

A wide index meets at least one of the following criteria:

- The index includes five or more columns.
- The index is more than 200 bytes.

Large indexes require more storage and increase the cost of index maintenance. If the index includes five or more columns because it is a covering index for multiple queries, the performance improvement might offset the additional overhead. However, if the index is **not** a covering index, the cost of maintaining the index could offset any performance improvement that the index provides. In this case, consider removing trailing edge columns.

## Table has no defined indexes

The table is being queried, but no indexes exist. If the table is very small, an index might not be more efficient than a full table scan. For larger tables, consider adding an index.

## Table does not have a primary key (PK)

A PK provides a way to uniquely identify a record and is necessary to ensure data integrity. A PK is also required to join the table. If no column or combination of columns provides a unique value, you can add an artificial PK such as an ID column.

# Identify blocking sessions and deadlocks with DPA

DPA provides information to help you determine if blocking sessions and deadlocks are affecting performance, and to investigate the root cause of these issues. See the following sections:

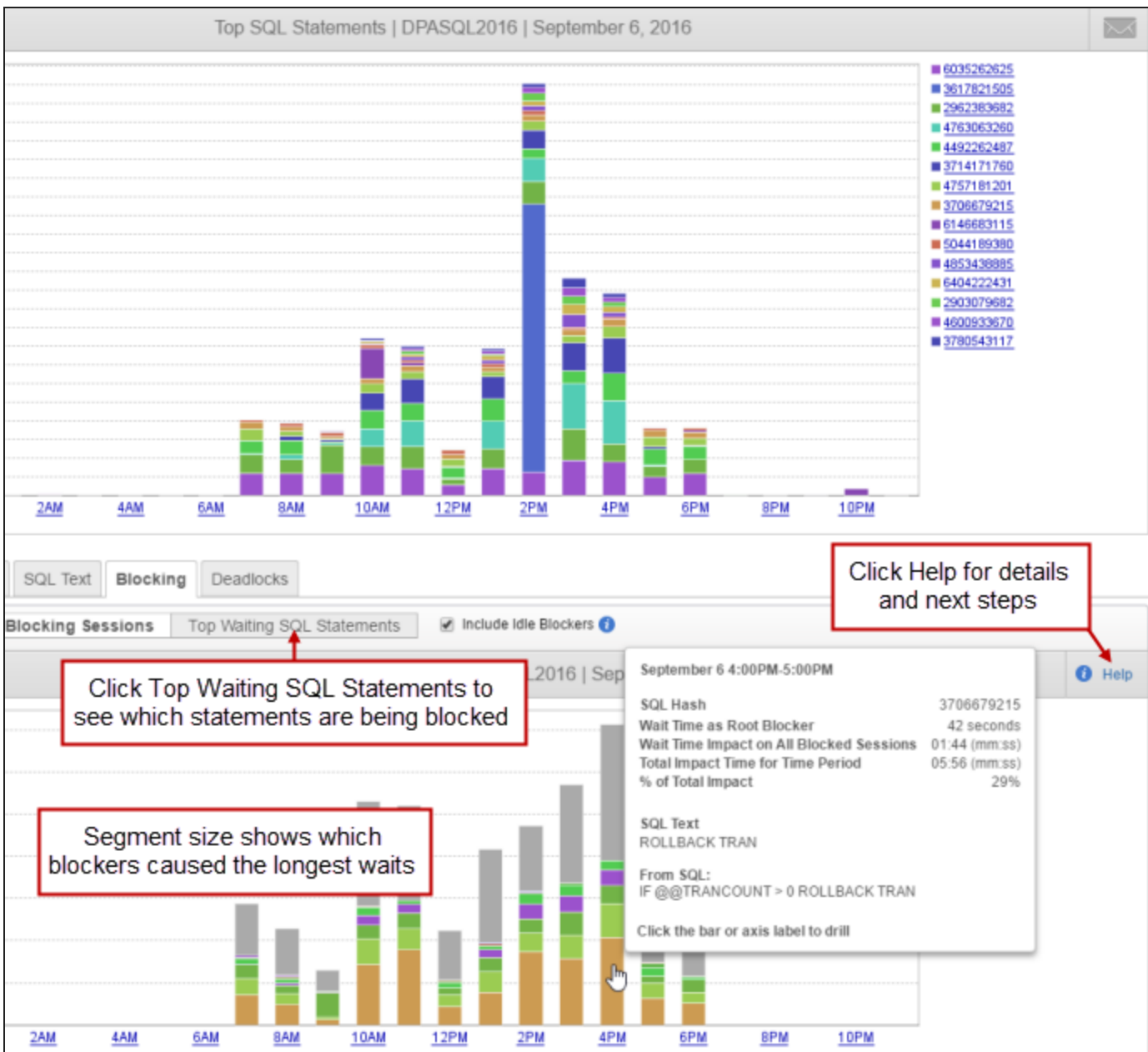
- [Identify blockers causing the longest waits](#)
- [Find the last activity of an idle blocker](#)
- [Investigate deadlocks on SQL Server instances](#)

## Identify blockers causing the longest waits

Are blocking sessions causing performance problems in your environment? Use the Blocking tab to identify the root blockers, find out which SQL statements are being blocked, and determine which blocking sessions are responsible for the longest overall waits. DPA shows the aggregated wait time for each blocker, which helps you focus your tuning efforts on blockers with the largest impact.

To view information about blocking sessions, click a database instance name on the DPA homepage to display the Trends charts. If necessary, click a bar on the chart to drill down to the time period you're interested in. Then click the Blocking tab below any Trends chart to view correlated information about blockers during that time period. The size of each segment in a bar provides a visual indicator of the waits that session caused.

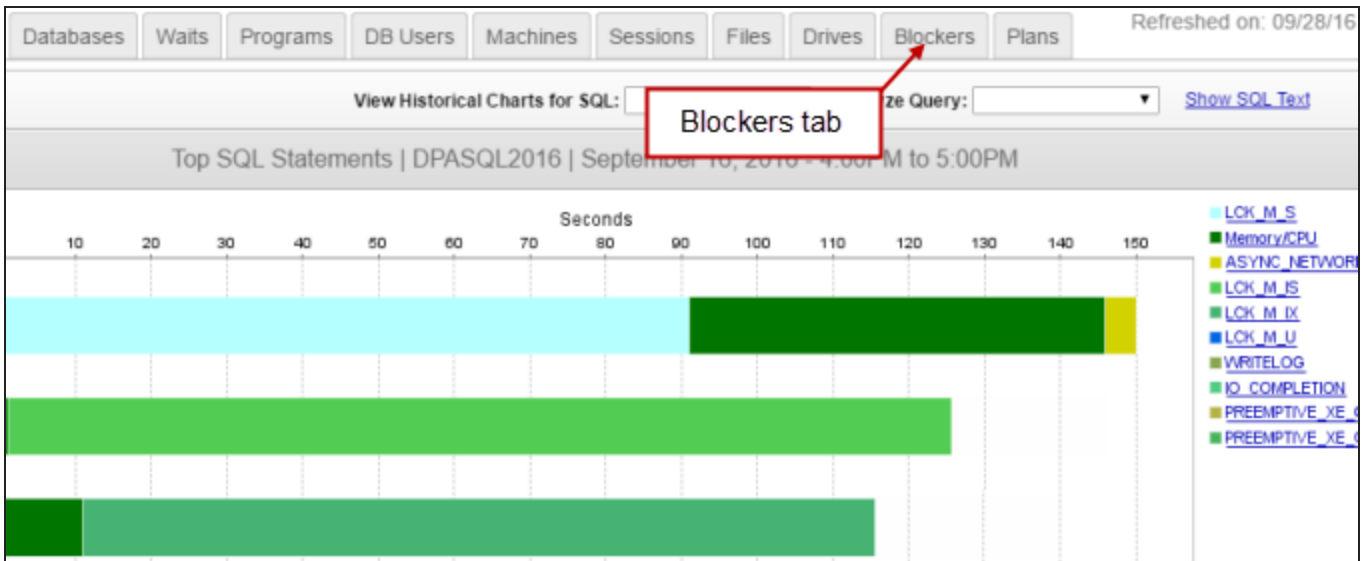




## Find the last activity of an idle blocker

Idle blockers can be difficult to diagnose because they are currently not performing any activity in the database. To help you find and fix the root problem, use DPA to determine what that session was doing before it became idle.

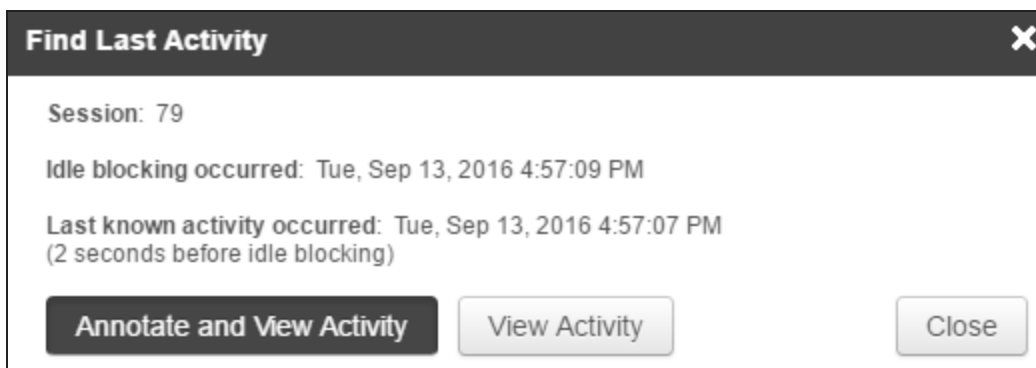
From the DPA homepage, click the name of a database instance. Click a bar on the Top SQL Statements chart to drill into a day, and then click a bar to drill into an hour. DPA displays information about the type of waits experienced during that hour.



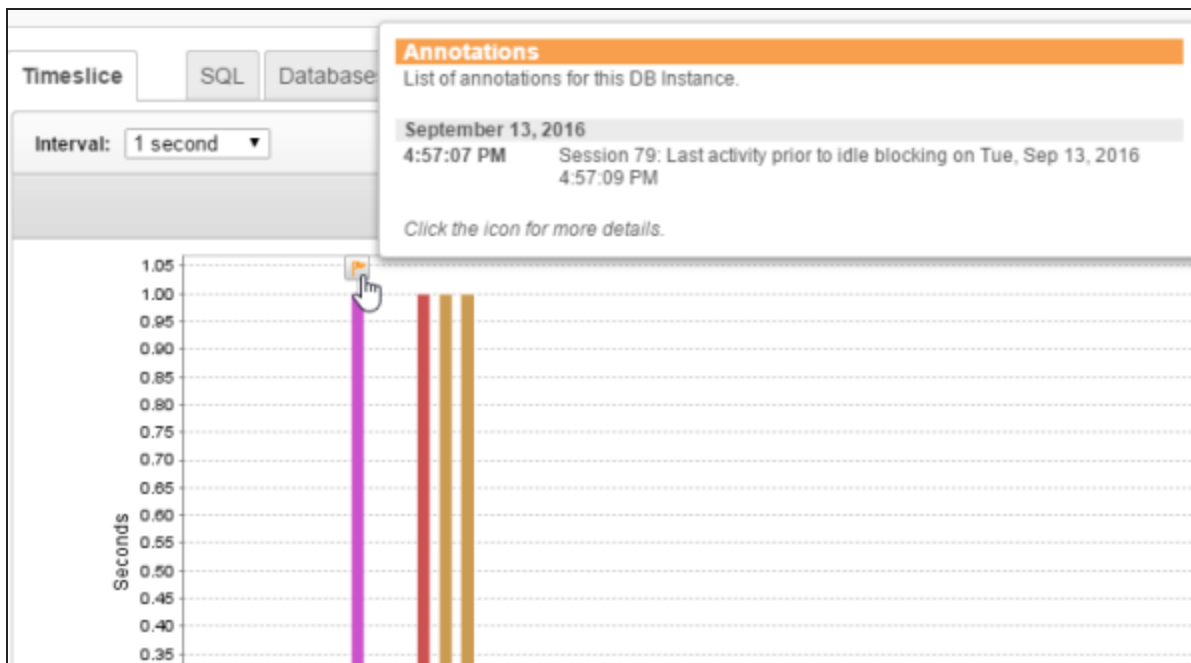
Click the Blockers tab above the chart to see a list of the blockers for that time period. You can expand a blocker to see information about the waits it caused. Each idle blocker row has a Find Last Activity link on the right.

		Blocking Time (seconds)					
SPID	Caused	Waited	User	Program	Machine	SQL	
81 (idle blocker)	104					<a href="#">Find Last Activity</a>	
87 (blocker and waiter)	30	4	swload	Order Entry	GIBSON	<a href="#">UPDATE address SET postal_cod</a>	
83 (waiter)		6	swload	Order Entry	GIBSON	<a href="#">UPDATE order_history_details SE</a>	
83 (blocker and waiter)		2	swload	Order Entry	GIBSON	<a href="#">UPDATE product SET price = price</a>	
72 (waiter)		5	swload	Accounting	acct-server	<a href="#">SELECT MAX(product_id) from pr</a>	
87 (waiter)		4	swload	Accounting	acct-server	<a href="#">UPDATE order_history_details SE</a>	
87 (blocker and waiter)	2	2	swload	Accounting	acct-server	<a href="#">UPDATE address SET phone = '9'</a>	
64 (waiter)		3	swload	Load Test	dev-bou-load	<a href="#">select * from address where city_i</a>	
62 (waiter)		2	swload	Order Entry	GIBSON	<a href="#">SELECT MAX(product_id) from pr</a>	
67 (waiter)		1	swload	Load Test	dev-bou-load	<a href="#">select * from address where city_i</a>	
83 (idle blocker)	33					<a href="#">Find Last Activity</a>	
81 (blocker)	31		swload	Load Test	dev-bou-load	<a href="#">Details</a>	

To find out what a blocking session was doing before it went idle, click Find Last Activity. The Find Last Activity dialog tells you when the last activity occurred.



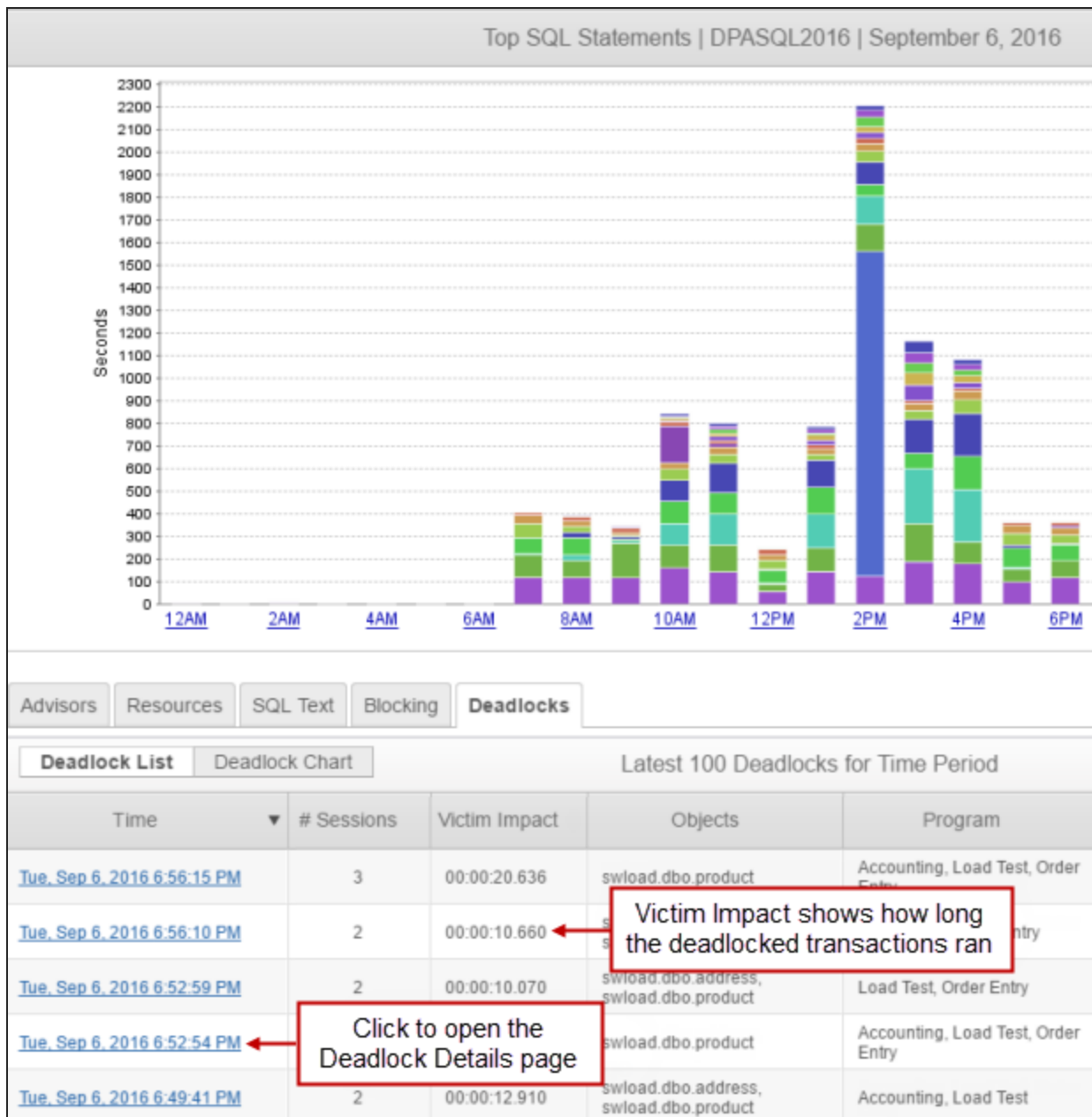
Click a button to view the activity. The Timeslice tab shows a bar representing the last SQL statement executed by the idle blocker. You can drill in to investigate further. If you clicked Annotate and View Activity, the SQL statement is automatically annotated to make it easy to find in the future.



## Investigate deadlocks on SQL Server instances

Deadlocks occur when two sessions have a lock on different resources, and each session needs the resource of the other to complete its task. One session (the victim) eventually releases its lock and does not complete its task. The transaction time that the victim spent in contention is a good measure of the impact that the deadlock had on performance.

For monitored SQL Server database instances, DPA provides detailed information about deadlocks, including the Victim Impact (how long the deadlocked transactions ran). Click the Deadlocks tab below any Trends chart to see the latest deadlocks for that time period.



Victim Impact shows how long the deadlocked transactions ran

Click to open the Deadlock Details page

Click the link in the Time column to open the Deadlock Details page, which includes the following sections:

- The Deadlock Summary section shows high-level information, including the Total Victim Impact.
- The Victims section shows details about the queries that were rolled back.
- The Survivor section shows details about the query that was completed.
- The Deadlocked Resources section shows the type of lock and the lock mode. Click the links for expert advice.

### Deadlock Summary

Deadlock Time:	Tuesday Sep 6, 2016 6:43:24 PM (DPA Time)
Number of Sessions:	3
Total Victim Impact:	00:00:18.127 (HH:MI:SS.mil)
Object:	swload.dbo.product
Programs:	Order Entry Accounting Load Test
User:	swload

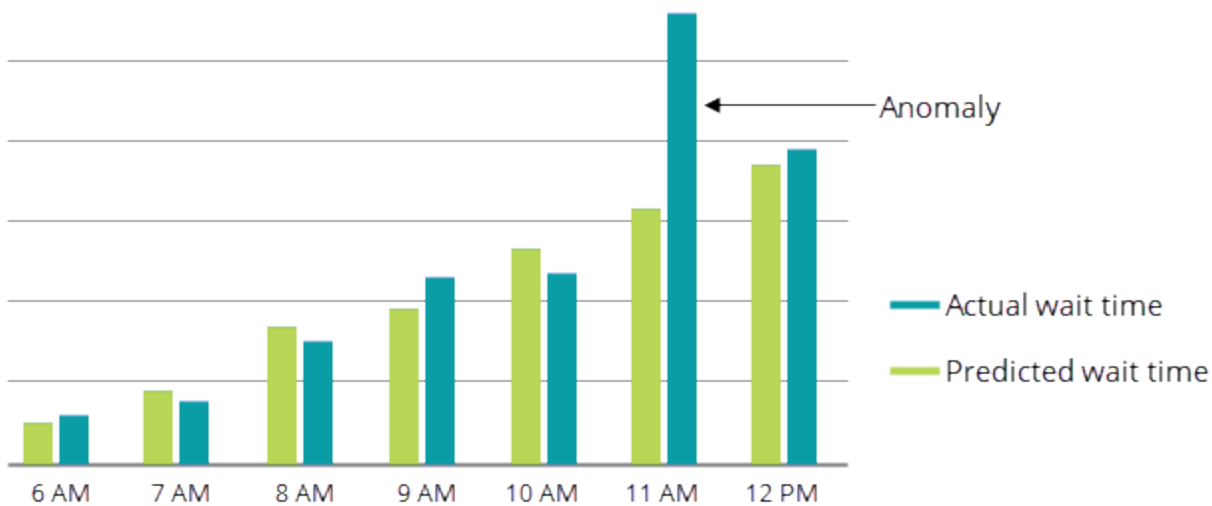
### Victims

SPID: **75**

Waiting on Object:	swload.dbo.product	SQL:	/* InputBuf */ UPDATE product SET price = price + .01
Program:	Order Entry		
User:	swload		/* Frame 1 procname=adhoc, line=1 */ unknown
Host:	GIBSON		
Isolation Level:	read committed (2)		/* Frame 2 procname=adhoc, line=1 */ unknown
Process ID:	processe8ec904e8		
Deadlock Wait Time:	00:00:03.618 (HH:MI:SS.mil)		
Transaction Time:	00:00:09.110 (HH:MI:SS.mil)		
Log Space Used:	0		
Server Batch ID:	0		

## Find and investigate unusually long wait times (anomalies)

DPA's [anomaly detection algorithm](#) identifies unexpected increases in wait time. DPA collects historical data and uses it to "learn" what normal is. DPA's proprietary algorithm makes predictions based on this data. When wait times for a time period are higher than expected, DPA reports an anomaly.



## Get notified when wait time is higher than expected

Configure the Database Instance Wait Time Anomaly alert to be notified whenever wait time is significantly higher than expected for a database instance. (To do this, [configure a Wait Time alert](#) and select Database Instance Wait Time Anomaly as the Alert Type.) This alert is triggered if the wait time for an instance was abnormally high during the most recently completed hour.



## View information about wait time anomalies

The wait time meter on the DPA home page indicates recently detected anomalies. Drill in to a database instance to view more detailed information on the Anomaly Detection charts.

### Wait time meter

On the DPA homepage, the wait time meter for each database instance provides information about recent database activity:

- The bar **length** shows the amount of wait time for each database instance as compared to all other monitored instances. Use the bar length to quickly identify instances with the highest wait times.
- The bar **color** identifies instances where DPA detected higher-than-expected wait times (anomalies). Yellow indicates a warning status, and red indicates a critical status. (For information about these thresholds, see [Anomaly thresholds](#).)


Monitor	Database Instance	Wait
↑ ON	[Instance Name]	<div style="border: 1px solid blue; padding: 2px;">           This instance had more wait time than any other, but the wait time was not higher than expected (green).         </div> 
↑ ON	[Instance Name]	<div style="border: 1px solid blue; padding: 2px;">           This instance had less wait time than the one above, but the wait time was significantly higher than expected (red).         </div> 

The wait time meter reflects recent database activity (a rolling one-hour time period). It is updated every 10 minutes to show the wait activity that occurred during the previous 60 minutes.

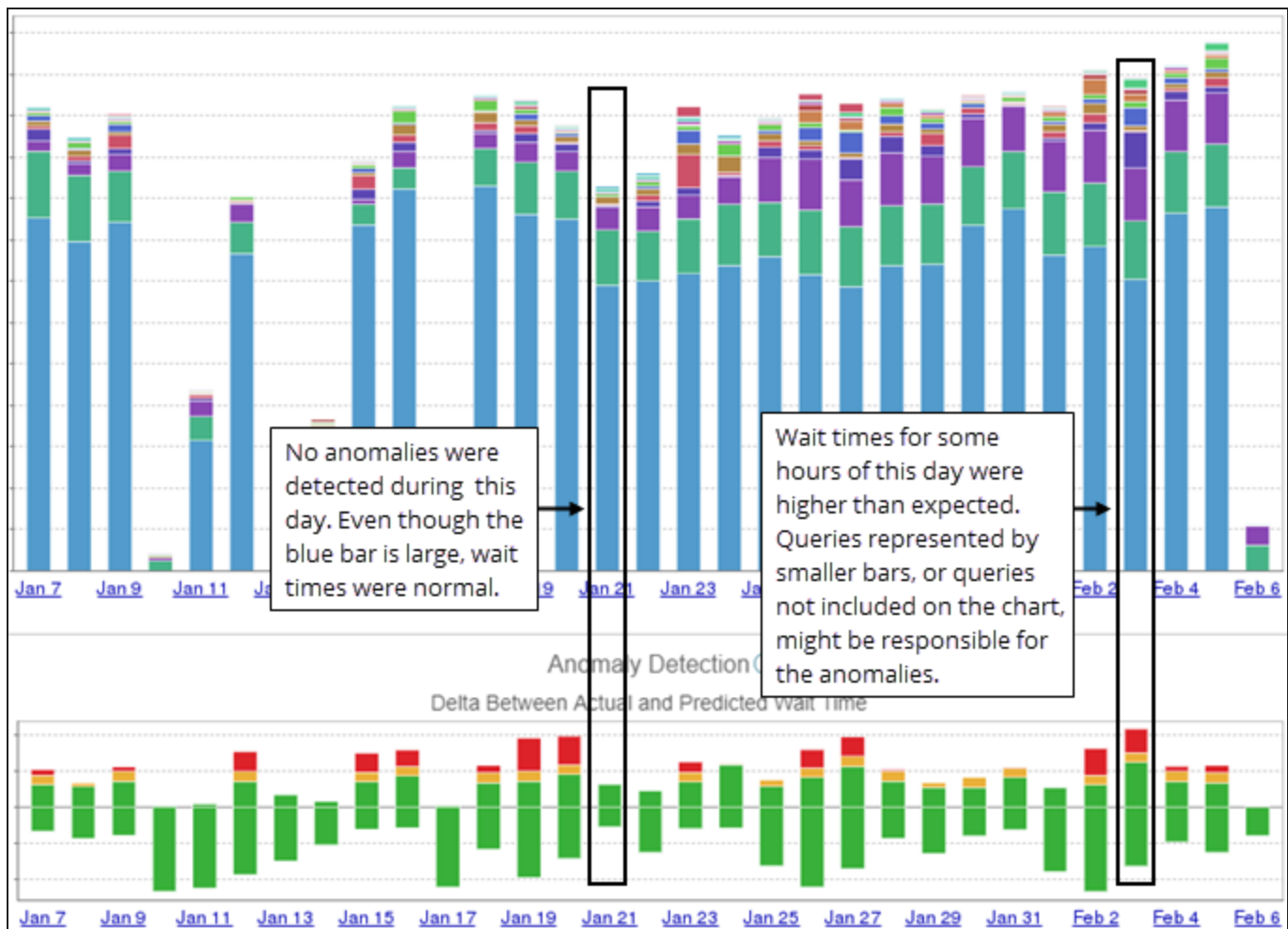
### Anomaly Detection chart (30-day period)

If DPA detects wait time anomalies for a database instance, click the database instance on the DPA homepage to drill in for more information. The Top SQL Statements chart and the Anomaly Detection chart show information from the past 30 days. These charts work together to help you understand the waits occurring in this database instance:

- The **Top SQL Statements chart** identifies the SQL statements with the highest wait times. In many cases, these are candidates for [tuning](#). But in other cases, further tuning is not possible or the wait times are not a problem. The large bars are normal, and you are more interested in unexpected increases in wait time.


 An anomaly is detected when the combined wait time for **all** SQL statements is higher than expected. The Top SQL Statements chart shows only the SQL statements with the highest waits, which might not be responsible for the anomaly.

- The **Anomaly Detection chart** identifies days when wait times were significantly higher than expected (wait time anomalies occurred).



Each bar on the Anomaly Detection chart shows a roll-up of the amount of wait time that the database instance experienced during that day.

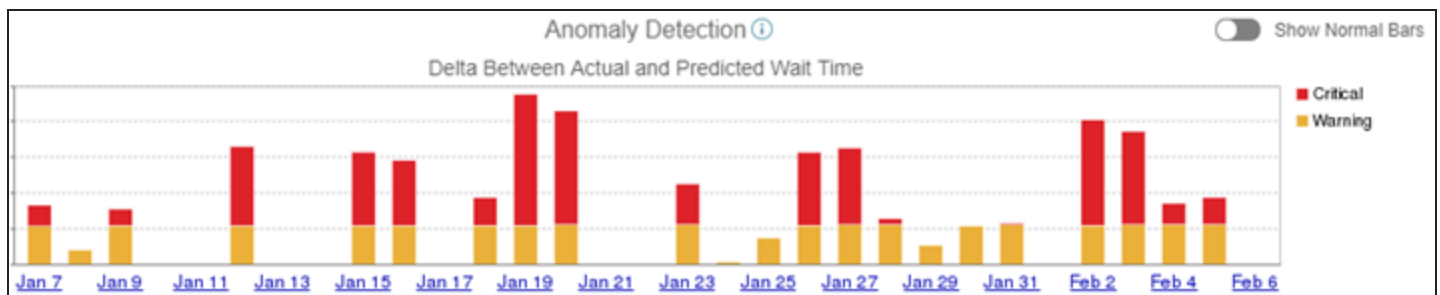
- Red segments indicate that wait times for one or more hours were much higher than expected (critical).
- Yellow segments indicate that wait times for one or more hours were higher than expected (warning).
- Green segments above the baseline (0) indicate that wait times for one or more hours were within the normal range, but slightly higher than expected.
- Green segments below the baseline indicate that wait times for one or more hours were lower than expected.

 DPA classifies all lower-than-expected wait times as normal, and does not alert on them.



Show only warning and critical segments

To focus only on segments that indicate wait time anomalies, you can deselect Show Normal Bars to hide the green bars.



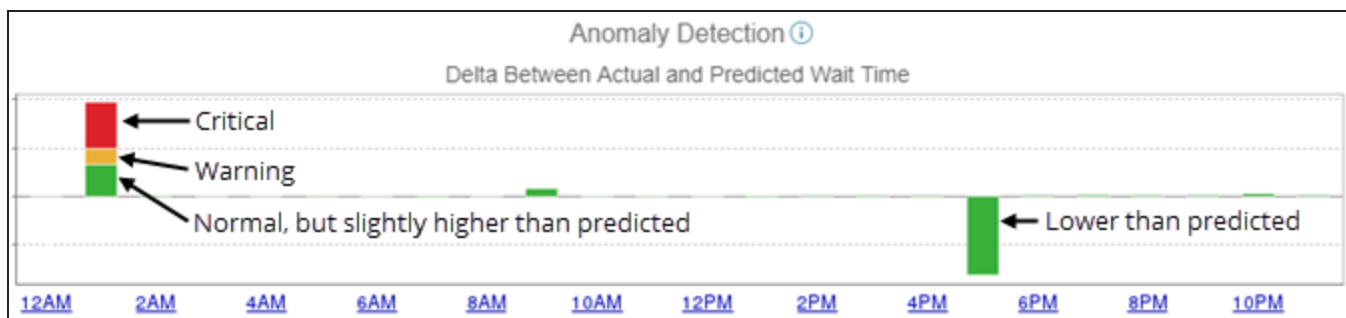
Drill in further

Click a bar that represents a day when anomalies occurred to display the Anomaly detection chart for that day.

### Anomaly Detection chart (one-day period)

The Anomaly Detection chart for a one-day period shows the differences between the predicted wait times and actual wait times for each hour. The bar for the current hour shows the differences during the six most recent 10-minute intervals (a rolling one-hour time period).

The baseline (0) represents the predicted value for the hour.



## Investigate higher-than-expected wait times

After you determine when anomalies are occurring, you can use either query performance analysis or DPA reports to help you determine which SQL statements are responsible for the anomalies.

### Determine when anomalies occurred

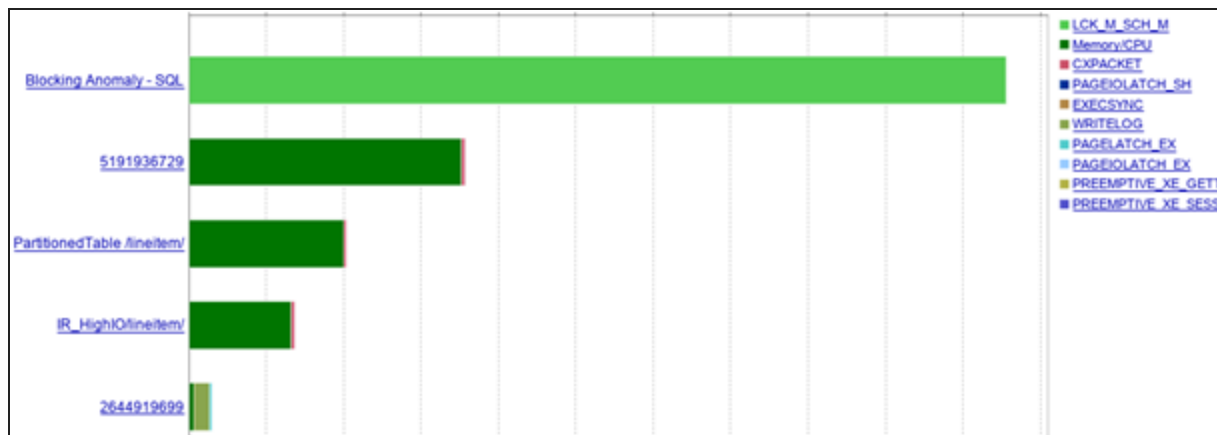
Use the Anomaly Detection charts to determine when anomalies occurred, and to see which SQL statements were running during that time period.

1. From the DPA homepage, click the database instance that is experiencing anomalies to display the 30-day Anomaly Detection chart.
2. Click a bar that represents a day when wait times were much higher than expected.

The one-day Anomaly Detection chart shows the hours when anomalies occurred. In this example, the 2 PM hour had the highest unexpected wait times.



3. Open the Anomaly Detection chart for a one-day period, and find the hours with large red segments. These are the hours when wait times were much higher than expected.
4. Click the bar that represents the hour, and view information about the SQL statements with high wait times that ran during that hour.



## Display historic wait times and performance analysis for these queries

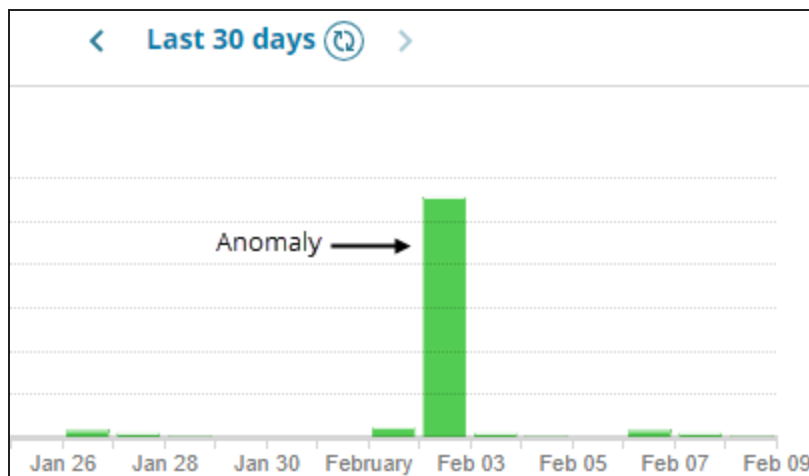
To determine which SQL statement is causing the anomaly, you can use the [Query Detail page](#) to view the historic wait times. It's usually a good idea to start with the bars at the top of the list. Also remember that more than one SQL statement might be causing the anomaly.

1. Click a bar that represents a SQL statement.

The Query Details page displays wait times for that SQL statement during the selected one-hour time period.

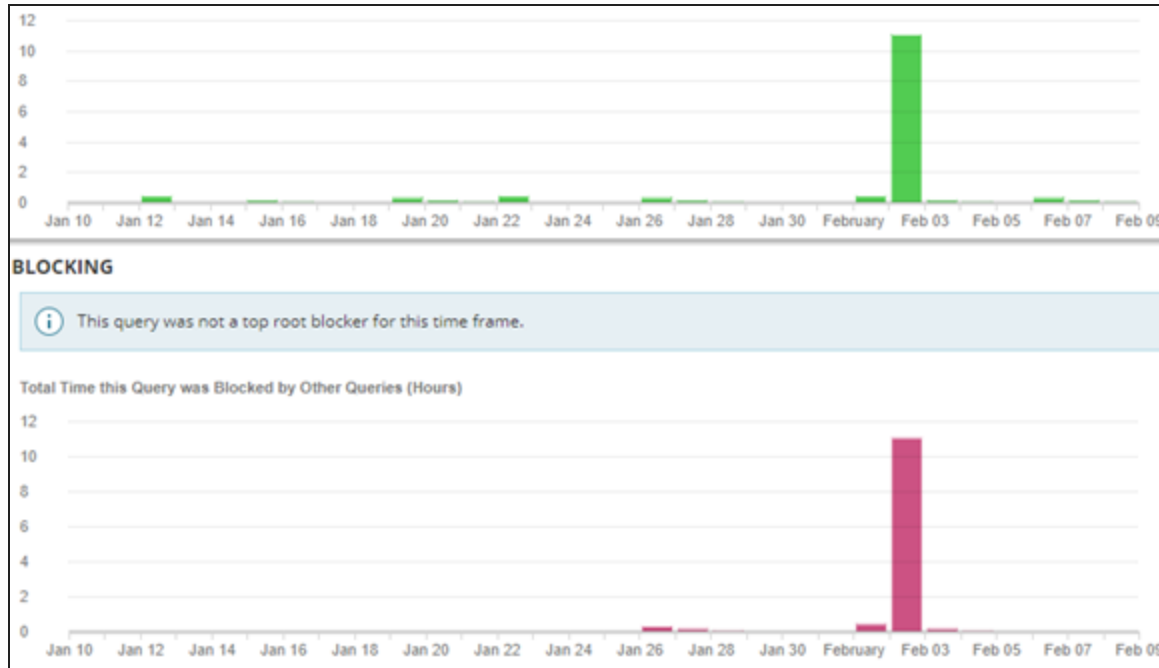
2. Click the time period at the top of the page and change the time range. For example, select Last 30 days or Last 90 days.

In this example, the wait times for February 2 are clearly an anomaly.



3. You can also scroll down to review DPA's query performance analysis for this SQL statement. In this example, we can see that the SQL statement was being blocked by other queries when

the anomaly occurred.



## Use DPA reports to review wait or query details

After you determine when the anomalies are occurring, you can create a report to review the wait times for that hour during the last 30 days to look for unusually high wait times.

1. Click Reports.
2. Select the database instance that is experiencing anomalies.
3. Select Top Waits as the Report Type.

Database Instance:	SQL2K17
Report Type:	Top Waits

4. Click Report Options.
5. Under Waits to Display, select the Top 50 Waits.

<input checked="" type="radio"/>	Top Waits Ranked by Cumulative Wait Time
Top	50 Waits

6. Under Dates to Display, select Last N Days as the Date Range, and leave 30 as the number of days.
7. Change the Hour Range to the time period when anomalies are occurring.

Date Range: Last N Days

Hour Range: 2:00pm to 3:00pm

Last: 30

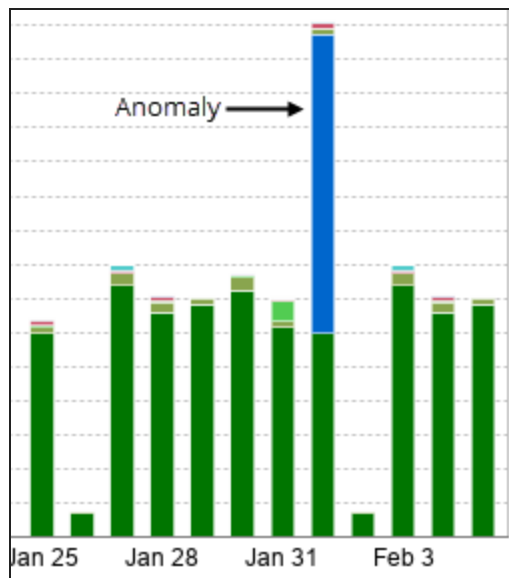
Days of Week: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday

Days Ending: Last Day Captured

Dates: January 7, 2019 - February 5, 2019

8. Click Display Report and review the wait times.

In this example, the anomaly stands out.



## Investigate lower-than-expected wait times

If wait times are much lower than expected, consider investigating to determine whether any SQL statements that normally run during that time period are missing.

1. Open the Anomaly Detection chart for a one-day period, and find the hour with the largest green segment below the baseline. Note the date and hour.
2. Click Reports.
3. Select the database instance, and select Top SQLs as the Report Type.
4. Click Report Options.
5. Under SQL Statements to Display, select the Top 50 SQL Statements.

Top SQL Statements Ranked by Cumulative Wait Time  
 Top  SQL Statements

6. Under Dates to Display, select Last N Days as the Date Range, and leave 30 as the number of days.
7. Change the Hour Range to the time period when wait time was much lower than expected.

Date Range:  Hour Range:  to

Last:  Days of Week:

Days Ending:

Dates: January 7, 2019 - February 5, 2019

8. Click Display Report and review the SQLs that ran each day to help determine if anything is missing.

## About anomaly detection in DPA

DPA uses an anomaly detection algorithm to determine if the wait times for a database instance are significantly higher than usual. In some cases, high wait times are normal and expected. With anomaly detection, DPA can alert you to unexpected increases in wait times, and help you [investigate these anomalies](#).

### How does DPA's anomaly detection work?

A machine learning algorithm uses wait time data that DPA collects to predict future wait times. DPA uses these predictions to detect wait times that are significantly higher than expected.

---

<b>Step 1:</b>	DPA gathers the data that the algorithm will use to learn what normal is and to predict future wait times. Up to 90 days of historical hourly data is used for learning.
<b>Data collection</b>	Anomaly detection requires a minimum of three days of learning data. DPA does not show any information about anomalies until it has collected at least three days of data. Predictions improve as more data is collected.

---

**Step 2:  
Data  
analysis  
and  
predictions**

Based on the learning data, the algorithm calculates:

- The amount of wait time that the database instance is likely to experience during each 1-hour period for the next 30 days.
- The standard deviation for the entire data set (which is used to calculate [thresholds](#)).

When enough data is available, predictions include daily and weekly seasonality (patterns of predictable fluctuations):

- Daily seasonality accounts for differences during each hour. For example, normal wait times at 2 AM are probably different than normal wait times at 2 PM.
- Weekly seasonality accounts for differences during each day of the week. For example, normal wait times at 2 PM on Saturday are probably different than normal wait times at 2 PM on Wednesday. (Weekly seasonality requires at least 30 days of learning data.)

**Step 3:  
Anomaly  
detection**

For each hour, DPA compares the actual amount of wait time during that hour to the predicted value. If the actual amount of wait time is above the warning or critical threshold, DPA:

- Changes the color of the wait time meter on the DPA homepage.
- Displays yellow or red segments on the bars in Anomaly Detection charts.
- Triggers the Database Instance Wait Time Anomaly alert, if it has been configured.

## How DPA determines the status of an incomplete hour

To determine if the wait time meter and hourly Anomaly Detection chart should show a warning or critical status for an incomplete hour, DPA uses the last 6 completed 10-minute intervals (a rolling one-hour interval). The status is updated every 10 minutes. For example, to determine the status of the 2:00 hour:

- From 2:00 to 2:09, DPA uses data from 1:00 to 1:59.
- From 2:10 to 2:19, DPA uses data from 1:10 to 2:09.
- From 2:20 to 2:29, DPA uses data from 1:20 to 2:19 (and so on).



For each 10-minute interval of the current hour, DPA uses a rolling one-hour interval to determine the status shown on the wait time meter. For example, 2:10 to 2:19 uses data from 1:10 to 2:09.

## SQL statements excluded from the trend charts

The anomaly detection algorithm uses the total wait time for the database instance, including wait time from any SQL statements that you have excluded from the trend charts. In most cases, a statement is excluded from the trend charts because it always has high wait times and the large bar dominates the charts. If the statement runs on a regular schedule with the expected amount of wait time, no anomaly would be detected during that time period, because high wait times are normal during that period. An anomaly would be detected only if wait times during that period were significantly higher than normal, in which case you might want to investigate the change.

## Does anomaly detection work well for all database instances?

DPA's anomaly detection algorithm, like most algorithms associated with workloads, works best when:

- The monitored database instances have a consistent workload executing against them.
- Daily and weekly seasonality is consistent. For example, database wait times are similar each Monday at 10 AM.
- DPA monitoring is always on (not shut down for hours or days at a time).

The algorithm might not work well when:

- The workload for a database instance is sporadic (for example, QA or reporting instances with inconsistent wait times).
- Daily and weekly seasonality is not consistent. For example, the workload on Monday at 10 AM varies from one week to the next, with no predictable pattern.
- DPA is not monitoring the instance consistently, and so it cannot get a good understanding of what normal is.

If anomaly detection does not work well for any of your monitored instances, SolarWinds recommends [disabling anomaly detection](#) for those instances.


## Large gaps in the learning data

If monitoring stops for more than 30 days, the anomaly detection algorithm does **not** make predictions based on the stale learning data collected before the 30-day gap. DPA collects new learning data and, after three days, begins to make predictions based on the current data.

## Anomaly thresholds

Anomalies are classified as warning and critical. The threshold for each classification is based on the standard deviation of the wait times for the associated time period.



 Standard deviation is a measure of how dispersed the values in a data set typically are.

The default values for the thresholds are listed below. You can [edit the associated configuration option](#) to change the default values.

Classification	Default Threshold	Configuration Option
Warning	The predicted wait time for the hour + 2 standard deviations	ANOMALY_DETECTION_THRESHOLD_WARNING
Critical	The predicted wait time for the hour + 3 standard deviations	ANOMALY_DETECTION_THRESHOLD_CRITICAL


## Specify the learning date after the load on a database instance changes

If the load on a database instance changes significantly (for example, because of changes in the network environment), the previously collected learning data is no longer accurate. To prevent this data from being used for anomaly detection, [set the advanced configuration option](#) `ANOMALY_DETECTION_FORCE_LEARNING_DATE` to the date when the load change occurred. Wait time data collected before this date will not be used to predict future wait times.

## Disable anomaly detection for a database instance

By default, anomaly detection is enabled for all database instances. To disable anomaly detection for a database instance that with an inconsistent workload or sporadic monitoring, [set the advanced configuration option](#) `ANOMALY_DETECTION_ENABLED` to `False` for that instance.


## Add an annotation to document a change to the database

 [Check out this video \(1:19\) on using annotations.](#)

When you make a change that could affect performance (such as adding an index, tuning a query, or adding resources), you can add an annotation in DPA to show when that change was made. The annotations are displayed on all trend and timeslice charts. By comparing performance data before and after the change, you can see what effect the change had.


1. From the DPA homepage, click the name of the database instance affected by the change.
2. Click Annotate in the upper-right corner of the trend chart.
3. Name the annotation, specify when it was added, and provide details about what change was

made and why.

 If your DPA server is in a different time zone, enter the DPA server time.

### Add Annotation

Annotation:

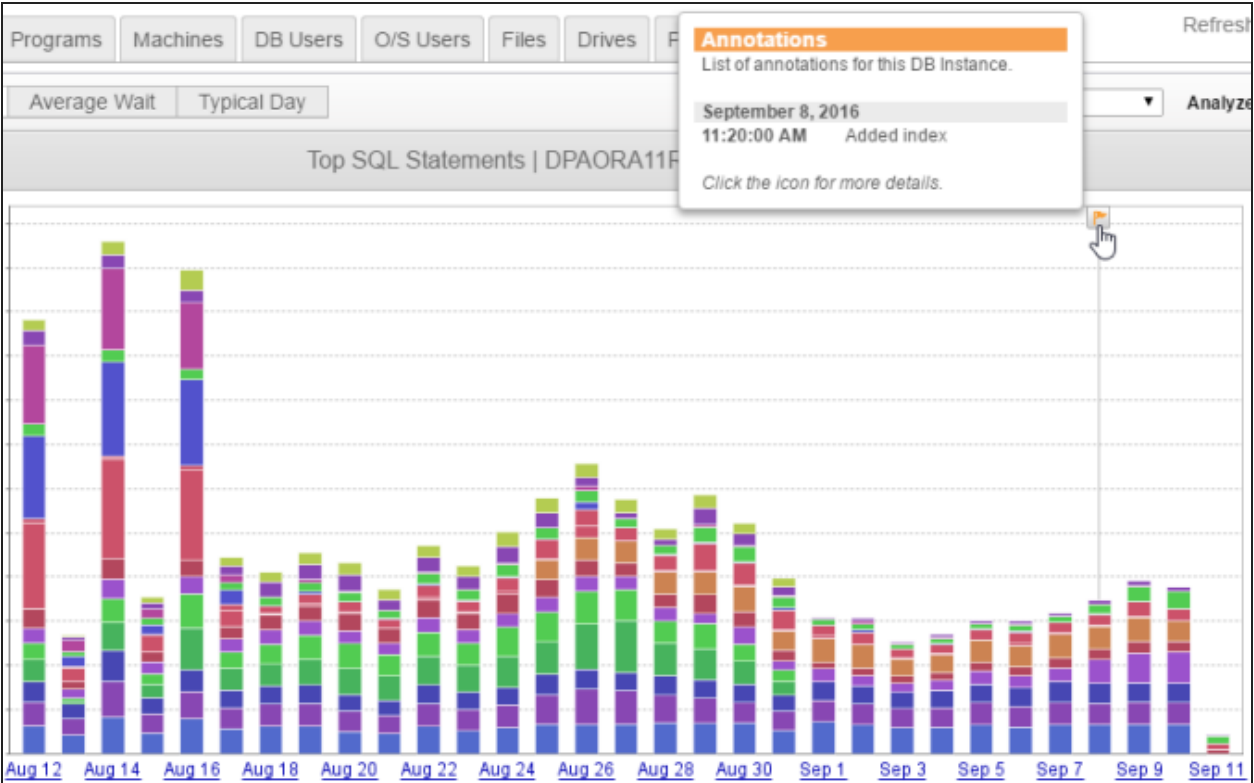
Occurred At:   Current DPA Time: 09/11/2016 1:46:23 PM -05:00

Details:

Created By:

4. Click Save.

The annotation is displayed as a flag on the chart. Point to the flag to see a summary, or click it to see details.



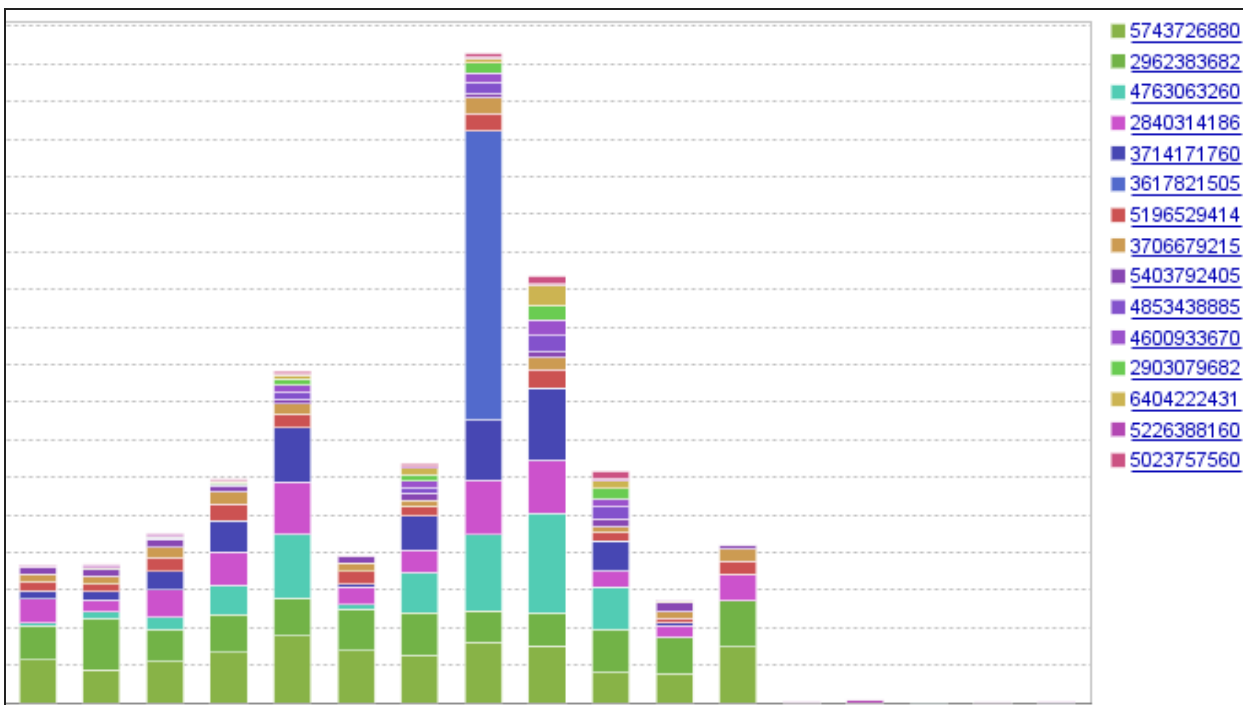
# Manage SQL statements

See the following topics to specify the properties of a SQL statement:

- [Name SQL statements](#)
- [Exclude SQL statements from DPA](#)
- [Add excluded SQL statements back to DPA trend charts and analysis](#)

## Name SQL statements

On the right side of each DPA trend chart, a legend identifies each SQL statement. By default, SQL statements are identified by their hash values.



When you are investigating a specific SQL statement, you can give it a name to make it easier to identify. The name appears in reports and chart legends.

1. In the chart legend, click the hash value that represents the SQL statement.  
The [Query Details page](#) displays information about the SQL statement.
2. In the top-right corner, click SQL Properties.
3. In the SQL Properties dialog, enter the name.

### SQL Properties (Hash: 1262681994, SQL ID: 33b6s4d5n5zwa) ✕

**SQL Name**

**Description** Optional

Add description to describe this query

**ADVANCED SETTINGS**

Show in Trends charts ?

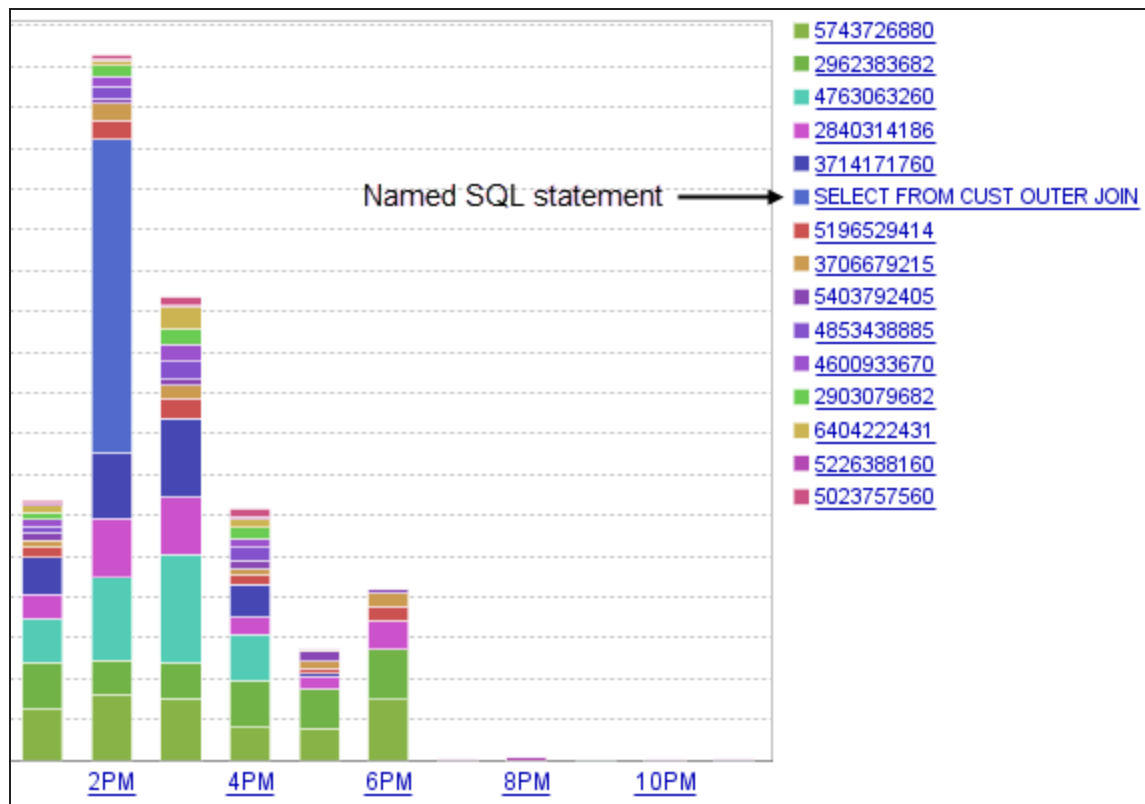
Enable advisor analysis ?

**SAVE** **CANCEL**

4. Click Save.
5. In the upper-left corner of the Query Detail page, click Back to return to the previous page.

The legend displays the name instead of the hash value.

? The colors representing the SQL statements on the chart might change after you name the SQL statement.



## Exclude SQL statements from DPA

Certain long-running SQL statements might not be candidates for tuning (for example, SQL statements associated with database backups, replication, or data loads). To prevent these statements from dominating trend charts or producing tuning advisors that are not actionable, you can exclude them from DPA.

**⚠** Before excluding SQL statements, consider the possible impacts. If an excluded SQL statement begins affecting your database performance, you will not see the issue in DPA because of the exclusion.

## Determine which option meets your needs

DPA provides three options for excluding SQL statements. Use this section to determine which option meets your needs, and then see the following sections for implementation instructions.

Summary	Option 1	Option 2	Option 3
Safe and easy	✓		
Excludes the SQL statement from both past and future charts and advisors	✓		
If the exclusion is reverted, DPA charts and advisors for the exclusion period show data about the excluded statement again	✓		
Excludes SQL statements from all DPA views and analysis, including reports and anomaly detection		✓	✓
Exclusion criteria is not restricted to the hash value or ID of one SQL statement			✓

### Option 1: Exclude a specific SQL statement from DPA charts and analysis

Use the SQL Properties dialog box to exclude the selected SQL statement from DPA trend charts, DPA tuning advisors, or both. When you exclude a statement from tuning advisors, DPA does not generate query advisors for it or consider it when generating table tuning advisors.

DPA recommends using this method if possible because it is safe (there is no risk of losing data) and easy (it is done through the interface).

- Data about the statement is **excluded** from:
- All DPA trend charts that represent one or more days, including charts that represent previous periods
  - All tuning advisors, including those generated for previous periods

- Data about the statement is still **included** in:
- Charts that represent less than one day
  - Reports
  - [Anomaly detection](#)

**i** If the statement runs on a regular schedule with a predictable amount of wait time, it would not cause DPA to detect an anomaly during the period when it runs. Higher wait times would be normal during that period.

If you **revert** the exclusion: DPA charts and advisors for the exclusion period show data about the excluded statement again. With this method, DPA continues to collect and store data about the SQL statement, and so the data is available.

## Option 2: Prevent DPA from storing data about a specific SQL statement


Run a statement against the DPA repository database that prevents DPA from storing the data that it collects about the specified SQL statement. The excluded SQL statement is identified by its DPA hash value.


This option for excluding SQL statements requires admin privileges.

Data about the statement is <b>excluded</b> from:	<ul style="list-style-type: none"> <li>Charts, tuning advisors, and reports that represent time periods <b>after</b> you ran the SQL statement</li> <li>Anomaly detection</li> </ul>
Data about the statement is still <b>included</b> in:	Charts, tuning advisors, and reports that represent periods <b>before</b> you ran the statement
If you <b>revert</b> the exclusion:	DPA charts and advisors for the exclusion period do not show data about the statement because that data is not stored in the DPA repository database.

## Option 3: Exclude SQL statements from collection based on criteria in the WHERE clause

Modify the WHERE clause of the DPA quickpoll query to prevent DPA from collecting information about SQL statements that meet certain criteria. With this method, you can exclude statements that come from a certain program, user, host, or a combination of factors. For example, you can exclude a specific SQL statement when it is run by a certain user, or you can exclude all SQL statements coming from a user on a certain computer.

 This method cannot be used for statements that run on a Sybase monitored database instance.

 In some cases, this method can improve the performance of the DPA quickpoll query on a busy database instance. However, adding too much logic to the WHERE clause can cause the query to run longer than expected or disrupt data collection.

If you use this option, DPA strongly recommends working with SolarWinds Support. Check CONTIME entries for QUICKPOLL\_EXECUTE both before and immediately after you apply the change to determine if there is any difference in performance.

Data about the statement is <b>excluded</b> from:	<ul style="list-style-type: none"> <li>Charts, tuning advisors, and reports that represent time periods <b>after</b> you modified the WHERE clause</li> <li>Anomaly detection</li> </ul>
---	--



Data about the statement is still **included** in:

Charts, tuning advisors, and reports that represent periods **before** you modified the WHERE clause

If you **revert** the exclusion:

DPA charts and advisors for the exclusion period do not show data about the statement because that data was never collected.

## Option 1: Exclude a specific SQL statement from DPA charts and analysis

1. In any chart legend, click the name or hash value that represents the SQL statement.


The [Query Details page](#) displays information about the SQL statement.

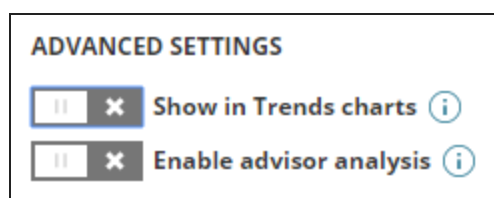
2. In the top-right corner, click SQL Properties.

The SQL Properties dialog opens.


3. Under Advanced Settings, clear one or both of the following options:

- Clear the Show in Trends charts setting to remove the SQL statement from multi-day or one-day trend charts. If you drill in to a time period less than one day, charts include the SQL statement.
- Clear the Enable advisor analysis setting to exclude this statement from the analysis that DPA runs to generate query advisors and table tuning advisors. When analysis is disabled, DPA does not detect problems with the SQL statement.

 When you clear the Show in Trends charts setting, both options are cleared. DPA does not perform analysis on SQL statements that are not shown in the Trends charts.



4. Click Save.

 You can [add excluded SQL statements back](#) to trend charts and analysis at any time.

## Option 2: Prevent DPA from storing data about a specific SQL statement

1. Log in to DPA using an account with admin privileges.
2. Get the hash value that identifies the SQL statement:
  - a. In any DPA chart legend, click the name or hash value that represents the SQL statement.  
The Query Details page displays information about the SQL statement.
  - b. In the top-right corner, click SQL Properties.
  - c. Copy the hash value from the top of the SQL Properties dialog box.
3. Open the DB query tool in DPA:
  - a. On the DPA menu, click Options.
  - b. Under Support > Utilities, click DB Query tool.
4. To get the database ID, enter the following query and click Execute Query:

```
select ID, name from cond;
```

The query returns the names and IDs of all monitored database instances.

**Enter Query**

Quick Query

Enter Queries - *Multiple queries can be entered when separated by a ';'.*

```
select ID, name from cond;
```

Execute query against:

---

Query Result - *(Repository) select ID, name from cond*

ID ▲	
1	DPASQL2K17LINUX
12	EVEREST@BOULDER

5. Enter the following query (replacing the variables with your database ID and SQL hash value), and click Execute Query:

```
INSERT INTO con_qp_exclude (dbid, type, value, origin) VALUES (databaseID, 'H', 'sqlHash', 'U');
```

All future data collection excludes this SQL statement. Past data is not purged, so DPA charts and reports that represent previous time periods will still include the SQL statement.

- i** To revert the exclusion, remove the SQL statement from the `con_qp_exclude` table by issuing a DELETE FROM statement:

```
DELETE FROM con_qp_exclude WHERE dbid=databaseID AND type='H' AND value='sqlHash';
```

## Option 3: Exclude SQL statements from collection based on criteria in the WHERE clause

1. On the DPA menu, click Options.
2. Under Administration > Configuration, click Advanced Options.  
The System Options tab lists options that apply to all database instances.
3. Click DB Instance Options and select the database instance on which the SQL statements run.
4. Select Support Options.
5. Click the name of the QUICKPOLL\_WHERE\_CLAUSE to open the Edit Option dialog.
6. Enter the phrase to include in the WHERE clause that specifies the SQL statements that should not be collected. Use the syntax appropriate for the database type. See the examples in the following sections.
7. Click Update, and then [restart DPA](#).

All future data collection excludes the SQL statements. Past data is not purged, so DPA charts and reports that represent previous time periods will still include the statements.

- i** To revert the exclusion, repeat this procedure to remove the criteria from the quickpoll WHERE clause.

### Examples for SQL Server

**Example 1:** Exclude all SQL statements from the TSQL program logging in from the server HPSEVER:

```
and not (s.program_name='TSQL' and s.hostname='HPSEVER')
```

**Example 2:** Exclude the specified SQL statement if it comes from a certain user, but do not exclude it if it comes from other users. Use the SQL\_handle (not the DPA hash value) to identify the SQL statement.

```
and not (s.loginname='Bob' and s.sql_handle=0x00987097097897)
```

**Example 3:** Exclude all SQL statements executed in the master database by the dataload program:

```
and not (db_name(s.dbid)='master' and s.program_name='dataload')
```

## Examples for Oracle

**Example 1:** Exclude all SQL statements from the SAP.exe program logging in from the server HPSEVER.

```
and not ("u".ksusepnm ='SAP.exe' and "u".ksusemnm='HPSEVER')
```

**Example 2:** Exclude the specified SQL statement if it comes from a certain user, but do not exclude it if it comes from other users.

```
and not ("u".ksuudlna='Bob' and "u".ksusesqh =97097897)
```

## Examples for Db2

**Example 1:** Exclude all SQL statements from the SAP.exe program logging in from the server HPSEVER:

```
and not (ai.appl_name ='SAP.exe' and ai.client_name='HPSEVER')
```

**Example 2:** Exclude a dynamic SQL statement if it comes from a certain user, but do not exclude it if it comes from other users.

```
and not (ai.auth_id='Bob' and s.stmt_text like '%insert into bad_table%')
```

**Example 3:** Exclude a static SQL statement if it comes from a certain user, but do not exclude it if it comes from other users.

```
and not (ai.auth_id='Bob' and st.text like '%insert into bad_table%')
```

## Examples for MySQL

**Example 1:** Exclude all SQL statements from the SAP.exe program logging in from the server HPSEVER:

```
and not (program_name = 'SAP.exe' AND host='HPSEVER')
```

**Example 2:** Exclude a dynamic SQL statement if it comes from a certain user, but do not exclude it if it comes from other users:

```
and not (user = 'BOB' AND statement_sql like '%insert into bad_table%')
```

## Examples for PostgreSQL

**Example 1:** Exclude the queries coming from a certain client address:

```
and a.client_addr != '10.140.66.28'
```

**Example 2:** Exclude a wait event type timeout that is coming from a client backend:

```
and a.wait_event_type != 'Timeout' and a.backend_type != 'client backend'
```


## Add excluded SQL statements back to DPA trend charts and analysis

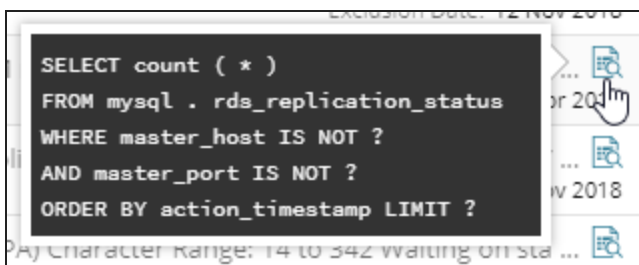
If you [excluded SQL statements](#) from DPA trend charts and analysis, you can add them back if needed.

### Add excluded SQL statements back to trend charts

1. On the DPA menu, click Options.
2. Under Administration > Configuration, click Excluded SQL Statements.

The Excluded SQL Statements dialog box lists the SQL statements that are excluded from trends charts.

3. Locate the SQL statement in the list:
  - Use the drop-down menu at the top to sort by exclusion date, database instance name, or SQL ID (name or hash).
  - Enter a string in the Search bar to show only SQL statements whose ID, database instance, or database type includes the search string.
  - Hold the mouse pointer over the  icon to display the SQL.



4. Select the SQL statement.
5. Click Re-include Selected.
6. Click the x in the top-right corner to close the dialog box.

## Add excluded SQL statements back to analysis

1. If the SQL statement is currently excluded from trends charts, add it back to trends charts.
2. In any chart legend, click the name or hash value that represents the SQL statement.

The [Query Details page](#) displays information about the SQL statement.

3. In the top-right corner, click SQL Properties.

The SQL Properties dialog opens.

4. Under Advanced Settings, select Enable advisor analysis.
5. Click Save.

# Resource metrics in DPA

Use the topics in this section to:

- [View resource metrics in DPA](#)
- [Show or hide VMware events](#) on metric charts
- View and learn about [resource metric baselines](#)
- [View or change DPA resource metric thresholds](#)
- Learn [what metrics DPA collects](#)

## View resource metrics in DPA

Resource metrics provide information about how resources (such as CPU, disk, and memory) are being used at specific points in time. These metrics show what was happening in the rest of your environment during database slow-downs, and can provide context to help you identify the root cause of performance problems.

DPA displays resource metrics in the following locations. You can:

- [View all available metrics on the Resources tab](#)
- [Correlate wait time with resource metrics on the Trends tab](#)
- [View resource metrics related to the performance of a query](#)

## View all available metrics on the Resources tab

The Resources tab displays all available resource metrics for the selected database instance.

1. On the DPA homepage, click a database instance to view detailed information.
2. In the upper-right corner, click the Resources tab.

The Resources tab displays all available resource metrics for the selected database instance.

You can:

- Click a time period in the upper-right corner to change the time range.
- Show or hide [baselines](#).
- Click the Information link to display information about a metric.
- Click Settings to [view or change the thresholds](#) for that metric.


Arrows at the top of each metrics chart indicate when VMware events occurred.

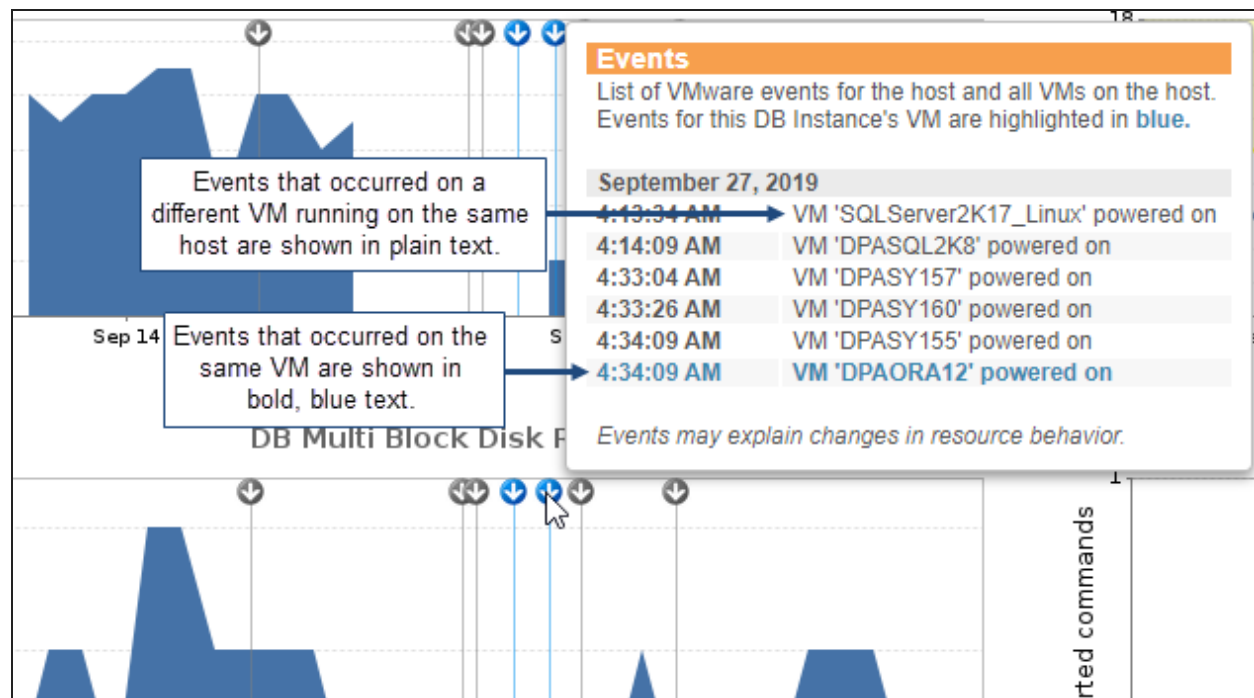
3. To view information about VMware events (for instances that run on a VM), hover over the blue

or gray arrows at the top of a chart.

Blue arrows indicate that events occurred on the VM where the database instance runs. Gray arrows indicate that events occurred on other VMs that run on the same host.

When you hover over an arrow, events that occurred on other VMs are shown in plain text. Those that occurred on the same VM are highlighted in bold, blue text.

 You can [choose which VMware events](#) to display on charts.





## Correlate wait time with resource metrics on the Trends tab

When you are viewing wait time charts on the Trends tab, you can scroll down to determine if unexpectedly long wait times correlate with resource contention.

1. On the DPA homepage, click a database instance to view detailed information.
2. In the upper-right corner, click the Trends tab.
3. Scroll down and click the Resources tab below the wait time charts.

The Resources tab displays a subset of the available resource metrics for the selected database instance. You can:

- Click Add Resource Chart to include additional charts.
- Click  to display information about a metric.
- Click  to [view or change the thresholds](#) for that metric.



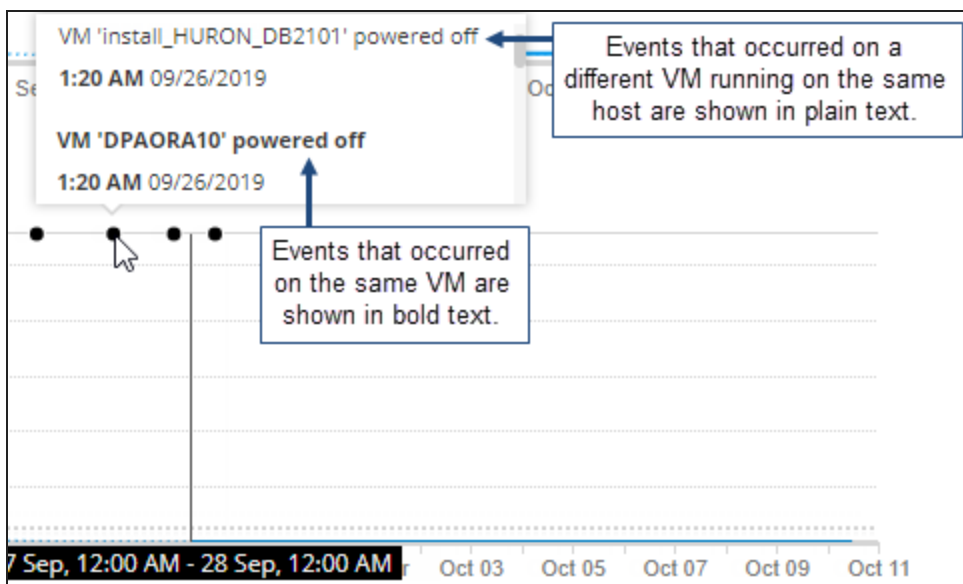
- Use other icons to the right of a chart to remove it, change its location, or replace it with a different chart.
4. To view information about VMware events that might explain changes in behavior, hover over the blue or gray arrows at the top of a chart. (See [step 3](#) in the previous section.)

 You can [choose which VMware events](#) to display on charts.

## View resource metrics related to the performance of a query

To help you find the root cause of long wait times for a query, the [Query Details page](#) includes the most relevant statistics, blocking, plan, and metrics charts. When you scroll down to view these charts, the Top Waits chart at the top of the page remains visible so you can correlate query wait times with other events during the same time period.

For instances that run on a VM, a black dot above a chart indicates when one or more VMware events occurred on that instance's VM or on a different VM running on the same host. Hover over the dot to display the list of VMware events, which might explain changes in behavior. Events that occurred on other VMs are shown in plain text. Those that occurred on the same VM are shown in bold text.



 You can [choose which VMware events](#) to display on charts.

## Show or hide VMware events on metric charts

For database instances that run on virtual machines (VMs), [resource metric charts](#) display VMware events. By default, the charts display events that occur on the VM where the database instance runs, as well as events on all other VMs that run on the same host. These events can sometimes explain changes in resource behavior.

- Events on the VM where the database instance runs are shown in bold text.
- Events on a different VM are shown in plain text.


You can change which VMware events DPA displays on metric charts. These settings apply when you open DPA on the current computer using the same type of browser (for example, Chrome).

1. On the DPA menu, click Options.
2. Under Administration > Display, click Display Options.
3. From the View Events drop-down menu, select which VMware events you want DPA to display on metric charts:
  - None
  - For this VM
  - For all VMs on host

## About DPA resource metric baselines

When you are viewing [resource metrics](#) on the Resources tab in DPA, you can display baselines to compare values from a specific period to historical norms. Baselines provide context for the current values. Metric values that are far above or below the baseline could indicate areas in need of tuning or reconfiguration.

Monitoring must be active for at least one day before baselines can be calculated, and baselines become more representative as more monitoring days pass.

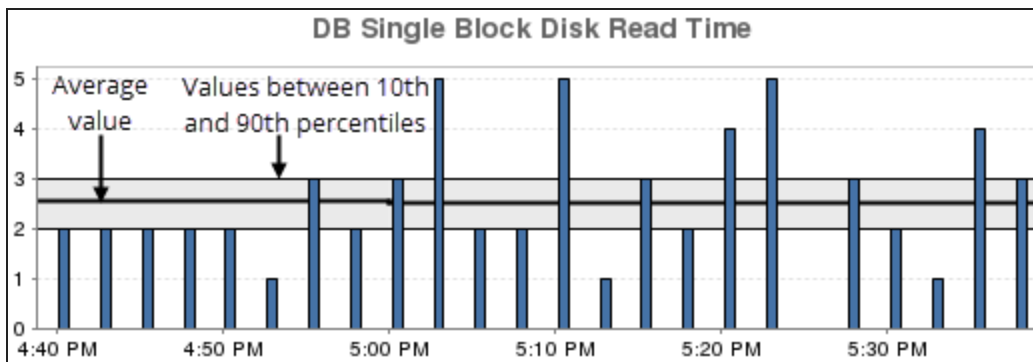
 Baselines are not available for metrics collected for the VM Option.

## Show or hide baselines

Baselines are available when the selected time period is one week or less.

Click the Show baseline or Hide baseline button near the top of the Resources tab to show or hide baselines. When you show baselines:

- A dark line represents the average value.
- If the time period is less than one week, a shaded area represents values between the 10th and 90th percentiles.



## How are baselines calculated?

Baselines are calculated for each one-hour period. By default, baselines are calculated using data only from weekdays (Monday through Friday). Each baseline is calculated using data from the corresponding hour for all weekdays, so the value for a specific hour is the same across all days. (For example, the value for 1 - 2 PM is the same Monday through Friday.)

Baselines are calculated using historical data from **before** the earliest time shown on the chart. For example, if a chart covers one week and starts on May 10th, all baselines are calculated using data from May 9th and earlier. For this reason, one-week charts show repeating patterns for each day.

## Change the days included in baseline calculations

To change the days included in baseline calculations, [update the advanced configuration option](#) `METRICS_BASELINE_TYPICAL_HOUR_CALCULATION`. This option can be set globally or for a specific monitored database instance.

Choose one of the following values:


Value	Description
Weekday Only (M-F)	Baselines are computed for each one-hour period using data from the corresponding hour on weekdays (Monday through Friday). This is the default.
All Days of the Week	Baselines are computed for each one-hour period using data from the corresponding hour on all days.
Same Day of Week	Baselines are computed for each one-hour period using data from the corresponding hour on the corresponding day. (For example, the value for 1 - 2 PM on Monday uses data from the corresponding hour on Mondays, and is therefore different than the value for 1 - 2 PM on Friday.) Be aware that this option increases the number of baselines per metric from 24 to 168.

## View or change DPA resource metric thresholds

[Resource metric charts](#) in DPA indicate when the metric has exceeded a Warning or Critical threshold. You can change the default thresholds to meet the needs of your environment. The custom thresholds can apply to a specific database instance or all monitored instances.

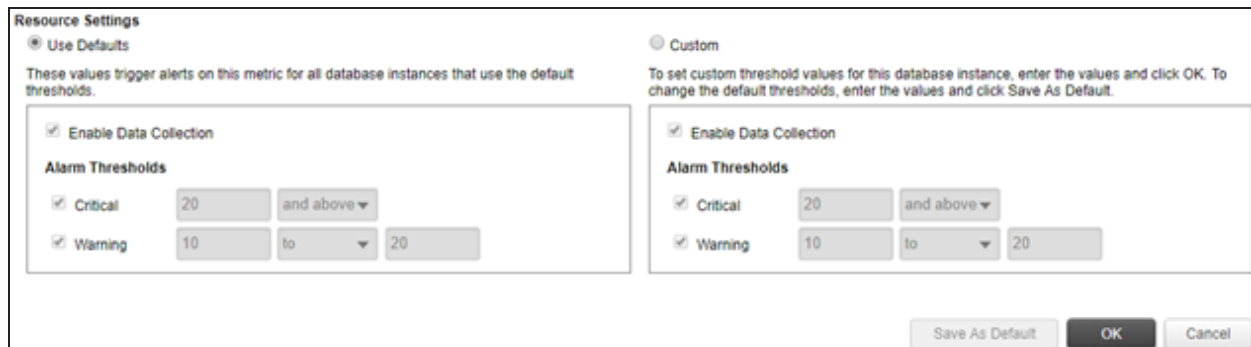
### View the current thresholds

1. On the DPA homepage, click the database instance whose resource metric thresholds you want to view.

 If you are going to change the default thresholds for all database instances, you can click any instance.

2. Click the Resources tab.
3. Click the tab that displays the metric whose thresholds you want to view or change.
4. Locate the metric chart and click Settings below the chart.

The Resource Settings page displays the current thresholds.



The screenshot shows the 'Resource Settings' dialog box. It has two main sections: 'Use Defaults' and 'Custom'. The 'Use Defaults' section is selected and contains a checkbox for 'Enable Data Collection' (checked), and 'Alarm Thresholds' with 'Critical' at 20 and above, and 'Warning' at 10 to 20. The 'Custom' section is unselected and contains the same 'Enable Data Collection' checkbox and 'Alarm Thresholds' settings. At the bottom right, there are three buttons: 'Save As Default', 'OK', and 'Cancel'.

### Change the thresholds

1. Select Custom.
2. Enter the new Critical and Warning threshold values.
3. Do one of the following:
  - To use the new values only for this database instance, click OK.
  - To use the new values for all database instances, click Save As Default. Then click Yes and OK at the confirmation prompts.

The new default threshold values are used for all database instances unless custom thresholds have been specified for an instance. Any database instance with Custom selected (instead of Use Defaults) will continue to use those custom thresholds.


## Metrics collected by DPA

The following topics describe the metrics that DPA collects:

- [Oracle metrics collected by DPA](#)
- [SQL Server metrics collected by DPA](#)
- [MySQL metrics collected by DPA](#)
- [Sybase metrics collected by DPA](#)
- [Db2 metrics collected by DPA](#)
- [Azure SQL database metrics collected by DPA](#)
- [ASMI metrics collected by DPA](#)
- [PostgreSQL metrics collected by DPA](#)
- [VM metrics collected by DPA](#)

## Oracle metrics collected by DPA

The following sections list the metrics that DPA collects for Oracle databases. Some metrics are not available for all Oracle deployments.

-  • Learn how to [view these metrics](#) and [change the thresholds](#).
- For detailed information about resolving issues, click the Information link next to the metric on the Resources tab. The Information link is not available for all metrics.

### CPU

Metric	Description
CPU Utilization by DB	The percentage of CPU being utilized by the database instance, which is a subset of the CPU utilized by the entire system. Oracle supplies this value only if the database parameter <code>timed_statistics = TRUE</code> . If this is high, use DPA Trends charts to review queries waiting on CPU.
O/S CPU Utilization	The percentage of CPU being utilized by the entire system. If this is high, compare this utilization with the CPU Utilization By Oracle metric. If most of the CPU is being used by Oracle, use the DPA Trends charts to review queries waiting on CPU. If Oracle is not using a significant portion of total CPU, review other non-Oracle programs running at this time.

## Memory

Metric	Description
Buffer Cache Hit Ratio	The rate at which this database finds the data blocks it needs in memory rather than having to read from disk. By itself, the buffer cache hit ratio is not very meaningful except for databases with undersized data buffer cache ( <code>db_cache_size</code> parameter). Oracle provides the data buffer cache advisory utility <code>v\$db_cache_advice</code> for assistance with sizing.
DB Logical Read Rate	The number of memory reads (session logical reads statistic from <code>v\$sysstat</code> ) per second for this database.
Shared Pool Size	The amount of memory allocated to the Oracle shared pool.
Buffer Cache Size	The amount of memory allocated to all Oracle buffer caches.
PGA Cache Size	The amount of memory allocated to the PGA cache.
Library Cache Hit Ratio	The library cache (a component of the shared pool) stores the executable (parsed or compiled) form of recently referenced SQL and PL/SQL code. Oracle tries to reuse this code. If the code has been executed previously and can be shared, Oracle will report a library cache hit. If Oracle is unable to use existing code, then a new executable version of the code must be built, which is known as a library cache miss.

## Disk

Metric	Description
DB Physical Read Rate	The number of kilobytes being read from disk every second for this database. If this is high, drill in to the DPA Trends charts and review queries waiting on physical read wait events (for example, <code>db file scattered read</code> or <code>db file sequential read</code> ).
DB Physical Write Rate	The number of kilobytes being written to disk every second for this database. If this is high, drill in to the DPA Trends charts and review queries waiting on write wait events (for example, <code>free buffer waits</code> or <code>direct path write temp</code> ).

Metric	Description
DB Physical I/O Rate	The number of kilobytes being read and written to disk every second for this database. If this is high, drill in to the DPA Trends charts and review physical read and write wait events.
DB Single Block Disk Read Time	The average number of milliseconds waiting for the <code>db file sequential read</code> event in this database. If this is high, contact your system administrator to understand why disk reads are slow. Use DPA to drill in to the <code>db file sequential read waits</code> and use the Files tab to show the disks involved.
DB Multi Block Disk Read Time	The average number of milliseconds waiting for the <code>db file scattered read</code> event in this database. If this is high, contact your system administrator to understand why these disk reads are slow. Use DPA to drill in to the <code>db file scattered read waits</code> and use the Files tab to show the disks involved.
DB Commit Time	The average number of milliseconds waiting for the <code>log file sync</code> event indicating commit times for this database.

## Network

Metric	Description
DB Round-trip Time	The round-trip time when running "select 1 from dual" (includes network time but not connect time) on this database. If this is high, contact your network administrator to understand network latency.
SQL*Net Sent Rate	The throughput of SQL*Net bytes sent to the clients in KB/second.
SQL*Net Received Rate	The throughput of SQL*Net bytes received from the clients in KB/second.

## Sessions

Metric	Description
DB Transaction Rate	The number of Transactions (user commits + user rollbacks statistics from v\$sysstat) being executed every second for this database.
DB Active Sessions	The number of sessions actively performing work or waiting for a resource (excludes idle sessions) for this database.
DB Blocked Sessions	The number of sessions that are blocked because another session is using a needed resource on this database.

## Waits

Metric	Description
Total Instance Wait Time	The total wait time for the database.

## License Compliance

Metric	Description
Connected Users	The number of distinct users connected to this instance (even if the connection is idle). This value is typically used for per-user licensing.
Connected Devices	The number of distinct client machines connected to this instance (even if the connection is idle). This value is typically used for per-device licensing. It can also be used to approximate per-user licensing.
Sessions	The number of sessions connected to this instance (even if the connection is idle). This value is typically used for licensing based on number of concurrent connections.
Core Count	The number of cores used by the instance. This value is typically used for per-core licensing.

## ASM

Metric	Description
ASM Summary Reads	The total number of all I/O read requests.
ASM Summary Writes	The total number of all I/O write requests.
ASM Summary Read Time	The average I/O time per read request over all disks.
ASM Summary Write Time	The average I/O time per write request over all disks.



Metric	Description
ASM Summary Write Rate	The total number of kilobytes written to disk every second.
ASM Summary Read Rate	The total number of kilobytes read from disk every second.

## Exadata

Metric	Description
IO Saved by Storage Cell Offloading	The amount of physical I/O that has been saved by offloading it to the Exadata storage servers. Each of the storage servers might get a piece of the SQL statement to operate on, so the processing is also parallelized at the same time. This saves valuable database server processing cycles for other non-I/O related activities and can dramatically reduce response times. Smart Scan is another term that essentially means the same thing.
Flash Cache Hit Ratio	The amount of I/O operations satisfied by the Exadata Smart Flash Cache within the Storage Servers. Exadata Smart Flash Cache is one of the essential technologies of the Oracle Exadata Database Machine that enables the processing of up to 1.5 million random I/O operations per second (IOPS), and the scanning of data within Exadata storage at up to 75 GB/second. This metric helps you understand how much the cache is helping.
Smart Scan Efficiency	When the storage cells process full table scans they can apply column filters and perform column projection so that not all blocks are returned to the database server, only the ones that are needed. This metric shows an efficiency of how well that is occurring. The data comes from <code>v\$sysstat</code> . It looks at the <code>'cell IO uncompressed bytes'</code> (a), <code>'cell physical IO bytes saved by storage index'</code> (b) and <code>'cell physical IO interconnect bytes returned by smart scan'</code> (c) metrics. It then applies the formula of $100 * (a + b) / c$ to get the percentage of data saved by the smart scans.
Cell Single Block Physical Read Latency	The average number of milliseconds waiting for the <code>cell single block physical read</code> Exadata event in this database.
Cell Multiblock Physical Read Latency	The average number of milliseconds waiting for the <code>cell multiblock physical read</code> Exadata event in this database.

Metric	Description
Cell Smart Table Scan Latency	The average number of milliseconds waiting for the <code>cell smart table scan</code> Exadata event in this database.


## RAC

Metric	Description
Avg GC CR Block Receive Time	<p>The average round-trip time or latency for all requests for a Consistent Read (CR) from this instance across the RAC Interconnect. If the transfer time is too high, or if one of the nodes in the cluster shows excessive transfer times, the RAC interconnect should be checked (using system level commands) to verify that it is functioning correctly. Calculation in (ms) is as follows:</p> $(gc\_current\_block\_receive\_time)/(gc\_cr\_blocks\_received) * 10$
Avg GC Current Block Receive Time	<p>The average round-trip time or latency for all processing requests for Current Mode Block from this instance across the RAC Interconnect. Calculation in (ms) is as follows:</p> $(gc\_current\_block\_receive\_time)/(gc\_current\_block\_receive\_time) * 10$
LMS Service Time	<p>The average LMS Service Time measures overall latency for a Consistent Read. This includes queue, build, flush, and send time. The Lock Manager Server (LMS) process, also called the GCS (Global Cache Services) process, is used to transport blocks across the nodes for cache-fusion requests. If there is a Consistent Read request, the LMS process rolls back the block, makes a Consistent Read image of the block, and then ships this block across the HSI (High Speed Interconnect) to the process requesting from a remote node. LMS must also check constantly with the LMD background process (or GES process) to get the lock requests placed by the LMD process.</p>
Current Block Service Time	<p>The average Current Block Service Time (ms) is calculated as follows:</p> $(gc\ current\ block\ pin\ time)+(gc\ current\ block\ flush\ time)+(gc\ current\ block\ send\ time)/(gc\ current\ blocks\ served) * 10$
Avg GC CR Block Build Time	<p>The average global cache CR block build times being experienced from this instance across the RAC Interconnect. The average time to build a consistent read block is calculated as follows:</p> $(gc\ cr\ block\ build\ time * 10)/(gc\ cr\ blocks\ served)$

Metric	Description
Avg Current Block Pin Time	The average current block pin times being experienced from this instance across the RAC Interconnect. This value is calculated as follows:  $(gc\_current\_block\_pin\_time * 10) / gc\_current\_blocks\_served$ as <code>average_pin_time</code>
Avg GC CR Block Send Time	The average global cache CR block send times being experienced from this instance across the RAC Interconnect. The average time to send a complete consistent read block is calculated as follows:  $(gc\_cr\_block\_send\_time * 10) / (gc\_cr\_blocks\_served)$
Avg Current Block Send Time	The average current block send times being experienced from this instance across the RAC Interconnect. This value is calculated as follows:  $(gc\_current\_block\_send\_time * 10) / gc\_current\_blocks\_served$ as <code>average_send_time</code>
Avg GC CR Block Flush Time	The average global cache CR block flush times being experienced from this instance across the RAC Interconnect. The average time spent waiting for a redo log flush is calculated as follows:  $(gc\_cr\_block\_flush\_time * 10) / (gc\_cr\_blocks\_served)$
Avg Current Block Flush Time	The average current block flush times being experienced from this instance across the RAC Interconnect. This value is calculated as follows:  $(gc\_current\_block\_flush\_time * 10) / (gc\_current\_blocks\_served)$

## SQL Server metrics collected by DPA

The following sections list the metrics that DPA collects for SQL Server databases.

-  Learn how to [view these metrics](#) and [change the thresholds](#).
- For detailed information about resolving issues, click the Information link next to the metric on the Resources tab. The Information link is not available for all metrics.

## CPU

Metric	Description
Signal Waits	<p>The percentage of total waits that are runnable and waiting for an available CPU. Anything over 20% indicates that there is a possible CPU resource bottleneck.</p> <p>Examine the overall wait events for the server as a whole. A high signal wait percentage could be due to an increased number of sessions, so examine the overall workload for the server as well. Take steps to either reduce the overall runtime for queries or reduce the total number of sessions.</p>
O/S CPU Queue Length via WMI	The number of O/S threads waiting to access the CPU for the entire system (includes all instances on this machine).
O/S CPU Utilization	The percentage of CPU being used for the entire system (includes all instances on this machine). Potential solutions to a CPU bottleneck are to reduce the server load by tuning the queries waiting on CPU, get faster CPUs, or get more CPUs.
Instance CPU Utilization	The CPU Utilization for this specific SQL Server instance. This is a subset of the O/S CPU Utilization metric.

## Memory

Metric	Description
O/S Memory Utilization	The percentage of memory being used for the entire system (includes all instances on this machine). If this is high and the Memory Paging Rate metric is high, you might need to increase the amount of physical RAM in the server, reduce the load on the server, or change the server memory configuration. Run <code>sp_configure</code> and review settings for "max server memory" and "min server memory" to determine amount of memory allocated to SQL Server.
Memory Paging Rate via WMI	The number of pages read from or to the disk to resolve memory references to pages that were not in memory at the time of the reference. This metric is for the entire system (includes all instances on this machine). High rates may indicate excessive memory contention (thrashing).

Metric	Description
Buffer Cache Hit Ratio	<p>The rate at which SQL Server finds the data blocks it needs in memory rather than having to read from disk for this instance. By itself, the buffer cache hit ratio is not very meaningful except for servers with undersized memory settings. Tuning queries and performing index optimization is the best way to increase buffer cache hit ratios. To see the current metrics for the buffer cache, run the following query:</p> <pre>select * from master..sysperfinfo where object_name like 'Buffer Manager'</pre>
Procedure Cache Hit Ratio	<p>The percentage of time when SQL Server looks for an execution plan in the procedure cache and finds it for this instance. If this is low, try to write more reusable code or consider increasing the size of the procedure cache. To see current metrics for the procedure cache, run the following query:</p> <pre>select * from master..sysperfinfo where object_name like '%Plan Cache%';</pre>
Page Life Expectancy	<p>The number of seconds a page will stay in the buffer pool without references. A lower value (for example, under 300) indicates the buffer pool is under memory pressure and you should add more memory to the system (enable AWE on 32-bit systems) or find the process in Task Manager that is consuming outside of SQL Server.</p>
Buffer Cache Size	<p>The current size of the SQL Server Buffer Cache.</p>
Plan Cache Size	<p>The current size of the SQL Server Plan Cache.</p>
SQL Compilations	<p>The number of compilations performed by SQL Server per second. Compilations are a natural part of SQL Server operations but do utilize CPU and other resources. Compare this to the Batch Requests/sec metric to understand if this metric is too high. Minimizing compilations will help overall performance. For more information, see the following Microsoft Knowledgebase article: <a href="http://support.microsoft.com/kb/243588">http://support.microsoft.com/kb/243588</a>.</p>
SQL Re-Compilations	<p>The number of re-compilations performed by SQL Server per second. Re-compilations occur for many reasons but this number should typically be low.</p>
Log Flushes	<p>The number of log flushes that occur per second.</p>
Log Bytes Flushed	<p>The number of bytes of information being flushed per second.</p>

## Disk

Metric	Description
Total I/O Wait Time	The sum of all I/O activity for all database files. If this is high: <ol style="list-style-type: none"> <li>1. Examine the current physical structure of databases on the server to see if it is possible to reduce I/O load by redistributing the database files to distinct disks.</li> <li>2. Examine queries and database design to determine if they can be tuned to reduce I/O.</li> </ol>
Total Read I/O Wait Time	The sum of all read I/O activity for all database files.
Total Write I/O Wait Time	The sum of all write I/O activity for all database files.
Physical Read Rate via WMI	The number of kilobytes being read from disk every second for the entire system (includes all instances on this machine). If this is high, drill in to the DPA Trends charts and review queries waiting on physical read wait types, such as <code>PAGEIOLATCH_SH</code> or <code>PAGEIOLATCH_EX</code> .
Physical Write Rate via WMI	The number of kilobytes being written to disk every second for the entire system (includes all instances on this machine). If this is high, drill in to the DPA Trends charts and review queries waiting on write wait types, such as <code>IO_COMPLETION</code> or <code>PAGEIOLATCH</code> .
Physical I/O Rate via WMI	The number of kilobytes being read and written to disk every second for the entire system (includes all instances on this machine). If this is high, drill in to the DPA Trends charts and review physical read and write wait types.
O/S Disk Queue Length via WMI	The number of I/O operations waiting for disk drives to become available for the entire system (includes all instances on this machine).
O/S Disk Queue Length	The number of I/O operations waiting for disk drives to become available for the entire system (includes all instances on this machine). Spikes of high disk queue length may be normal, but if this is high for an extended period, you could have an I/O bottleneck. Drill in to DPA Trends charts to examine queries with I/O wait types during the timeframe.

Metric	Description
SQL Disk Read Latency	Disk read latency from <code>dm_io_virtual_file_stats</code> DMO.
SQL Disk Write Latency	Disk write latency from <code>dm_io_virtual_file_stats</code> DMO.
Page Reads	The number of SQL Server physical reads from disk to memory. OLTP workloads are typically about 80-90 per second with higher values (or spikes) being an indication of insufficient storage performance, insufficient indexing, or not enough memory.
Page Writes	The number of SQL Server physical writes from memory to disk. OLTP workloads are typically about 80-90 per second. If this is high (or spikes) it needs to be cross checked with lazy-writes/sec and checkpoints in order to determine if the issue might be due to low memory.

## Network

Metric	Description
Round-trip Time	The round-trip time when running "select 1" against this instance (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

## Sessions

Metric	Description
Transaction Rate	The number of transactions being executed every second in this instance (the Transactions/sec statistic from <code>sysperfinfo</code> for the instance).
Blocked Sessions	The number of sessions that are blocked in this instance because another session is using a needed resource.
Active Sessions	The number of sessions in this instance actively performing work or waiting for a resource (excludes idle sessions).
Batch Requests	The number of batches being executed by SQL Server every second.

## Waits


Metric	Description
Total Instance Wait Time	The total wait time for the instance.

## License Compliance

Metric	Description
Connected Users	The number of distinct users (that is, login names) connected to this instance (even if the connection is idle). This value is typically used for per-user licensing.
Connected Devices	The number of distinct client machines connected to this instance (even if the connection is idle). This value is typically used for per-device licensing. It can also be used to approximate per-user licensing.
Sessions	The number of sessions connected to this instance (even if the connection is idle). This value is typically used for licensing based on number of concurrent connections.
Core Count	The number of cores used by the instance. This value is typically used for per-core licensing.


## MySQL metrics collected by DPA

The following sections list the metrics that DPA collects for MySQL database instances.

-  Learn how to [view these metrics](#) and [change the thresholds](#).
- For detailed information about resolving issues, click the Information link next to the metric on the Resources tab. The Information link is not available for all metrics.




## Memory

Metric	Description
InnoDB Buffer Pool Hit Ratio	<p>The rate at which the InnoDB engine finds the data blocks it needs in memory rather than having to read from disk. <code>Innodb_buffer_pool_size</code> is a very important parameter for InnoDB performance. A rule of thumb is to set the <code>innodb_buffer_pool_size</code> up to the total size of the database but not to exceed about 70% of total RAM.</p> <div data-bbox="284 478 1513 577"><p> If you have MyISAM tables, you want to balance the <code>key_buffer_size</code> and the <code>innodb_buffer_pool_size</code> to best utilize memory for your MySQL instance.</p></div> <p>If the hit ratio is lower than 90%, investigate increasing the buffer pool in <code>my.cnf</code> and <code>my.ini</code> by updating the <code>innodb_buffer_pool_size</code> system variable and then restarting MySQL.</p>
InnoDB Buffer Pool Consumed Space	<p>The percentage of the InnoDB buffer pool that contains data. <code>Innodb_buffer_pool_size</code> is a very important parameter for InnoDB performance. A rule of thumb is to set the <code>innodb_buffer_pool_size</code> up to the total size of the database but not to exceed about 70% of total RAM. A general good practice is to size the buffer pool such that it is mostly full. By doing this, it indicates that you are not wasting memory and that queries are finding the majority of their data in the buffer pool.</p> <p>If this metric is either too high or too low, consider the following:</p> <ul style="list-style-type: none"><li>• If the Consumed Space is consistently low, this indicates that your buffer pool is too big and memory is unnecessarily allocated to the buffer pool. Investigate lowering the <code>innodb_buffer_pool_size</code> variable.</li><li>• If the Consumed Space is consistently very high (99% or higher), this may indicate that the size of the buffer pool is too low. Check the resource metric InnoDB Buffer Pool Hit Ratio. If this metric is periodically or consistently low, investigate increasing the <code>innodb_buffer_pool_size</code> variable.</li><li>• If the Consumed Space is low, but it is on the rise, this indicates that the buffer is being initially populated with data. No action is needed at this point.</li></ul>

Metric	Description
InnoDB Buffer Pool Flushed Page Rate	<p>The number of requests per second to flush pages from the InnoDB buffer pool to the data file. Flushing pages to disk is a normal InnoDB operation. InnoDB tries to do this activity in the background when the total load is low.</p> <p>If the flush rate is too high, consider the following:</p> <ul style="list-style-type: none"> <li>• If the InnoDB log files are too small, this forces a checkpoint operation that flushes buffer pool pages to disk. Check the InnoDB Log Write Rate metric. If you see a lot of log writes that correspond to high InnoDB Buffer Pool Flushed Page Rate values, increase the <code>innodb_log_file_size</code> variable.</li> <li>• A buffer pool size that is too small can cause frequent flushes. A rule of thumb is to set the <code>innodb_buffer_pool_size</code> up to the total size of the database but not to exceed about 70% of total RAM.</li> </ul> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>i</b> If you have MyISAM tables, balance the <code>key_buffer_size</code> and <code>innodb_buffer_pool_size</code> values to best utilize memory for your MySQL instance.</p> </div> <ul style="list-style-type: none"> <li>• Check the load, mostly writes, on the system and investigate ways to decrease the load. Although SELECTs can also cause pages to be flushed from the buffer pool to disk, writes usually cause higher flush rates.</li> <li>• Optimize SQL to reduce the number of rows being written:             <ol style="list-style-type: none"> <li>1. Go to the Trends page for the timeframe and look at the UPDATE, DELETE, and INSERT statements with the highest wait time.</li> <li>2. Determine which statements have the highest Rows Affected or Sent value on the SQL Data tab.</li> <li>3. For these statements, consider evaluating WHERE clauses to ensure you process only rows that are required.</li> </ol> </li> </ul>
InnoDB Buffer Pool Data Pages	The number of pages that contain data in the InnoDB buffer pool. This includes both dirty and clean pages.
InnoDB % of Dirty Buffer Pool Pages	The percentage of InnoDB buffer pool data pages that have been changed in memory but have not yet been written (flushed) to disk.

## Disk

Metric	Description
InnoDB fsync Call Rate	The number of InnoDB fsync() system calls per second made to flush both the data and log files to disk.
InnoDB Log Write Rate	<p>The number of requests per second to write to the InnoDB redo log. The general recommendation is to set the combined size of log files to about 25%-100% of the buffer pool size to avoid unnecessary buffer pool flush activity on log file overwrite.</p> <p> A larger log file size will increase the time needed for a recovery process.</p> <p>If this is one of your top metrics, consider increasing the <code>innodb_log_file_size</code> in <code>my.cnf</code> and <code>my.ini</code> and then restarting MySQL.</p>
InnoDB Data Read Ops Rate	The number of InnoDB data read operations per second.
InnoDB Data Write Ops Rate	The number of InnoDB data write operations per second.

## Network

Metric	Description
Bytes Sent	<p>Throughput of bytes sent from MySQL to clients. If Bytes Sent has an abnormal spike or if it is higher than normal, consider the following:</p> <ol style="list-style-type: none"> <li>1. Examine the Bytes Received (KB/s) metric together with Bytes Sent (KB/s) to gain a more complete story of network traffic.</li> <li>2. Optimize SQL to reduce network traffic. Go to the Trends page to identify which SQL statements have the highest wait time. Determine which of these statements have the highest Rows Affected or Sent statistic on the SQL Data tab. For these statements: <ul style="list-style-type: none"> <li>• Evaluate WHERE clauses to ensure you are processing only rows that are required.</li> <li>• Eliminate columns from your result set that you don't need.</li> <li>• Use summary tables where possible to limit the number of rows processed/returned.</li> <li>• Rewrite complicated queries to assist in processing fewer rows.</li> </ul> </li> <li>3. Enlist the assistance of your network admin to evaluate network traffic, with a focus on the traffic between the Application server and the MySQL server.</li> </ol>
Bytes Received	<p>Throughput of bytes received by MySQL from clients. If Bytes Received has an abnormal spike or if it is higher than normal in general, consider:</p> <ol style="list-style-type: none"> <li>1. Examine the Bytes Received (KB/s) metric together with Bytes Sent (KB/s) to gain a more complete story of network traffic.</li> <li>2. Check the LOAD DATA infile statements which can contribute to the network traffic.</li> <li>3. Enlist the assistance of your network admin to evaluate network traffic, with a focus on the traffic between the Application server and the MySQL server.</li> </ol>
Round-trip Time	<p>The round-trip time when running "select 1" against this instance (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.</p>

## Sessions

Metric	Description
Blocked Threads	The number of threads that are blocked because another thread is holding a lock on an object, typically a table or an index. Drill down in the Trend page to locate additional details about what the blocking sessions are doing. Tune the queries you find by adding indexes or rewriting queries to minimize the time the locks are held.

Metric	Description
Active Threads	<p data-bbox="289 220 1510 430">The number of active threads in the database instance to support client connections. This metric is based on the MySQL Global Status variable <code>threads_running</code>. MySQL associates each client connection with a dedicated thread that handles all requests for that connection. This means that there are as many threads as there are clients currently connected.</p> <p data-bbox="289 451 1510 745">MySQL employs a thread cache to reduce the performance penalties associated with creating and destroying threads. The size of the thread cache is governed by the <code>thread_cache_size</code> system variable. When a connection is established, MySQL creates a new thread if an available thread cannot be found in the thread cache. When a connection ends, its thread is returned to the thread cache unless the cache is full. To monitor how many threads are being created because no cached thread is available, look at the Created Threads (sessions) metric.</p> <p data-bbox="289 766 1510 976">Each thread has some overhead in the form of server and kernel resources, including stack space, that affects the ability to scale to handle large numbers of connections. If you need to handle a large number of simultaneous connections, a common solution is to decrease the thread stack size. Doing so will limit memory-consuming activities conducted by the thread.</p> <p data-bbox="289 997 1023 1039">If Active Threads is too high, consider the following:</p> <ul data-bbox="332 1060 1510 1669" style="list-style-type: none"><li data-bbox="332 1060 1510 1144">• Use connection pooling in your applications to reduce the number of simultaneous queries.</li><li data-bbox="332 1155 1510 1239">• Use the MySQL master/slave architecture and move some or all SELECT queries to a slave.</li><li data-bbox="332 1249 1510 1501">• MySQL may be incurring excess overhead such as memory. If you feel that this is a problem, you can decrease the thread stack size, but you need to realize that this will limit memory-consuming activities conducted by the thread. In other words, it limits complexity of SQL statements and stored program recursion depth. To set the stack size, start the server with <code>--thread_stack=N</code> where N is in bytes.</li><li data-bbox="332 1512 1510 1669">• If the Active Threads value is higher than the thread cache size, MySQL may be incurring excess expense due to the creation and destruction of threads. If you feel that this overhead is a problem, you can try increasing the size of the thread cache by increasing the value of the <code>thread_cache_size</code> system variable.</li></ul>

Metric	Description
Created Threads	<p>The number of created threads in the database instance to support client connections in the given interval. This metric is based on the MySQL Global Status variable <code>threads_created</code>. MySQL associates each client connection with a dedicated thread that handles all requests for that connection. This means that there are as many threads as there are clients currently connected.</p> <p>Because thread creation and disposal can be expensive, MySQL employs a thread cache. When a connection is established, MySQL will create a new thread if an available thread cannot be found in the thread cache. When a connection ends, its thread is returned to the thread cache unless the cache is full. To monitor how many threads are currently running (cached or not), look at the Active Threads (sessions) metric.</p> <ul style="list-style-type: none"><li>• If the Thread Creation Rate value is high and the thread cache is not full, this generally means that the cache is being filled, which is a normal situation. To see how many threads are in the cache and the size of the thread cache, look at the <code>threads_cached</code> and <code>thread_cache_size</code> system variables.</li><li>• If the Thread Creation Rate value is high and the thread cache is full, this means that available threads are not being found in the thread cache, causing new connections to create new threads, which can be an expensive operation. If you think this overhead is causing problems, you can try increasing the size of the thread cache by increasing the value of the <code>thread_cache_size</code> system variable.</li><li>• Consider using connection pooling in your application(s).</li></ul>
Connection Attempts	<p>The number of connection attempts in the given time interval (successful or not). This metric is based on the MySQL Global Status variable <code>connections</code>.</p> <p>If the Connection Attempts value is high, investigate the connection attempts in the logs. Enable logging of the connection attempts in the following ways:</p> <ul style="list-style-type: none"><li>• If you are only interested in aborted attempts, make sure that the value (level) of the <code>log_warning</code> system variable is 2, and then check the error log.</li><li>• If you are interested in successful and aborted connections, make sure that the general query log is enabled by checking the <code>general_log</code> system variable. The location of the general query log file is in the <code>general_log_file</code> system variable. Enabling the general query log can decrease the performance of the MySQL server, as every connection attempt and SQL statement will be logged.</li></ul>

## Waits

Metric	Description
Total Instance Wait Time	The total wait time for the instance.

## InnoDB Logical I/O

Metric	Description
InnoDB Row Read Rate	<p>The number of rows that are read from InnoDB tables per second. An occasional spike in this rate can indicate that a mysqldump backup task is running.</p> <p>If you see a high InnoDB Row Read Rate that you believe is contributing to slow performance, consider optimizing the SQL to reduce the number of rows being read:</p> <ol style="list-style-type: none"> <li>1. Go to the Trends page for the time frame and look at the statements with the highest wait time.</li> <li>2. Determine which of the statements have the highest 'Rows Examined' value on the SQL Data tab.</li> <li>3. For these statements, consider the following: <ul style="list-style-type: none"> <li>• Use summary tables where possible to limit the number of rows processed.</li> <li>• Rewrite complicated queries to assist in processing fewer rows.</li> <li>• Evaluate WHERE clauses to ensure you process only rows that are required</li> </ul> </li> </ol>
InnoDB Buffer Pool Read Rate	The number of logical read requests per second from the InnoDB buffer pool. High values usually indicate high load on the system. Reads from the buffer pool are efficient reads, so high rates only rarely indicate a performance problem.
InnoDB Buffer Pool Write Rate	The number of requests per second to write to the InnoDB buffer pool.



## Objects

Metric	Description
Table Cache Hit Ratio	<p>The percentage of time that MySQL used an available cached "file descriptor" (that is, an <code>.frm</code> file that contains a table's underlying format).</p> <p>Whenever MySQL needs to access a table, it needs the table structure. The structures of previously opened tables are stored in the table cache. If a table's structure has not been cached, MySQL needs to load the structure from disk into cache, negatively affecting database performance. The lower this ratio is, the more the database has to load table structures from disk. Table structures are stored in <code>.frm</code> files on disk (<code>tableName.frm</code>).</p> <p>If the Table Cache Hit Ratio is low, increase the <code>table_open_cache</code> variable in <code>my.cnf</code> and <code>my.ini</code>. Recommendations:</p> <ul style="list-style-type: none"><li>• Set <code>table_open_cache</code> to the total number of tables in the database.</li><li>• A typical range for the <code>table_open_cache</code> is from 2000 (default) to 100,000.</li></ul>
Table Cache Filled	<p>The percentage of the cache that is filled with "file descriptors" (that is, an <code>.frm</code> file that contains a table's underlying format).</p>

## Sorts/Joins

Metric	Description
Row Sort Rate	<p>The number of rows sorted per second while executing statements. If MySQL cannot use an index to retrieve presorted rows, it performs a sort that increments the <code>sort_rows</code> counter.</p> <p>If this metric is high, consider these solutions:</p> <ul style="list-style-type: none"><li>• Check the Sort Merge Passes resource metric and determine if there is a need to increase <code>sort_buffer_size</code>.</li><li>• Optimize the SQL to reduce sorting.<ol style="list-style-type: none"><li>1. Go to the Trends page for the time frame and look at the statements with the highest wait time.</li><li>2. Find the statements with the highest Rows Sorted value on the SQL Data tab.</li><li>3. For these statements, consider using a combined or covered index with the same columns in the same order as the ORDER BY clause. In some cases, MySQL can use an index to satisfy an ORDER BY clause without performing any extra sorting.</li></ol></li></ul>

Metric	Description
Sort Merge Passes	<p>The number of merge passes per second performed by the sort algorithm. A Sort Merge Pass occurs if sorting large amounts of data using a limited amount of space. Performance suffers when these sorts can not be performed in memory. When the sort buffer overflows, MySQL creates temporary files on disk to use in the file sorting and merging algorithm. The data is sorted in multiple passes to first sort small chunks of data before merging the results together.</p> <ul style="list-style-type: none"><li>• Look at the Row Sort Rate metric. If there is a lot of sorting happening in this time frame, follow the suggested solutions.</li><li>• Increase the global <code>sort_buffer_size</code> system variable to improve the performance of queries that sort a lot of data.</li><li>• Increase the <code>sort_buffer_size</code> at the session level with a SET statement. Add the statement to your application code before running these kinds of queries. For example: <code>SET session sort_buffer_size = 8M</code></li><li>• Use an index with columns in the ORDER BY clause. MySQL might use this index to satisfy an ORDER BY clause without extra sorting.</li></ul>
Joins By Table Scan	<p>The number of joins that performed table scans (that is, joins that did not use indexes).</p>

Metric	Description
Temp Table Creation Rate	<p>The number of internal temporary tables created per second while executing statements. MySQL creates internal temporary tables to process operations such as SELECT ... GROUP BY / ORDER BY and SELECT DISTINCT. Unfortunately, temporary tables larger than the sizes specified in <code>tmp_table_size</code> and <code>max_heap_table_size</code> have to be converted to a slow, disk-based MyISAM temporary table. Likewise, if the query uses TEXT or BLOB fields, MySQL always has to use slow, disk-based temporary tables because in-memory temporary tables don't support those fields.</p> <p>If this metric is high, you run the risk of temporary tables being created on disk. Consider the following:</p> <ol style="list-style-type: none"><li>1. Go to the Trends page to identify which SQL statements have the highest wait time.</li><li>2. Find the statements with the highest Temp Tables Created statistic on the SQL Data tab.</li><li>3. For these statements:<ul style="list-style-type: none"><li>• Use a combined or covered index that has the same columns in the same order as the ORDER BY clause. In some cases, MySQL can use an index to satisfy an ORDER BY clause without doing any extra sorting.</li><li>• Remove TEXT/BLOB fields if they are not needed for the query.</li></ul></li></ol> <p>Consider also increasing the <code>tmp_table_size</code> or <code>max_heap_table_size</code> values to reduce the number of internal temporary tables that have to be written to disk.</p>
On-Disk Temp Table Creation Rate	<p>The number of internal on-disk temporary tables created per second while executing statements.</p>

## Statements

Metric	Description
Statements Execution Rate	<p>The number of statements executed per second, not including those executed from stored programs. This is only a problem if your users complain about poor performance. Consider the following:</p> <ul style="list-style-type: none"> <li>• Identify and tune the queries with the highest wait time.</li> <li>• Look to see if there are a high number of executions of a SQL Statement. Look for possible ways to modify your application to decrease the number of executions, such as caching.</li> <li>• If you believe poor performance is due to the volume of statements being executed, consider implementing a MySQL master and slave architecture and move some or all SELECT queries to a slave.</li> </ul>
Statements Execution Rate from Stored Programs	<p>The number of statements executed per second from programs. This is only a problem if your users complain about poor performance. Consider these measures:</p> <ul style="list-style-type: none"> <li>• Identify and tune the queries with the highest wait time.</li> <li>• If you believe poor performance is due to the volume of statements being executed, consider implementing a MySQL master and slave architecture and move some or all SELECT queries to a slave.</li> </ul>
Select Statement Rate	The number of times a SELECT statement has been executed per second.
Insert Statement Rate	The number of times an INSERT statement has been executed per second.
Update Statement Rate	The number of times an UPDATE statement has been executed per second.
Delete Statement Rate	The number of times a DELETE statement has been executed per second.

## Sybase metrics collected by DPA

The following sections list the metrics that DPA collects for Sybase database instances.

- Learn how to [view these metrics](#) and [change the thresholds](#).
- For detailed information about resolving issues, click the Information link next to the metric on the Resources tab. The Information link is not available for all metrics.

## CPU

Metric	Description
CPU Utilization By Sybase	The percentage of CPU being used by the database instance, which is a subset of the CPU used by the entire system. If this is high, use DPA's Trends charts to review queries with the wait type "waiting on run queue after sleep".

## Memory

Metric	Description
Buffer Cache Hit Ratio	<p>The rate at which Sybase finds the data blocks it needs in memory rather than having to read from disk.</p> <p>By itself, the buffer cache hit ratio is not very meaningful except for servers with undersized memory settings.</p> <p>Tuning queries and performing index optimization is the best way to increase buffer cache hit ratios.</p> <p>To determine the current sizes of the data caches, use the <code>sp_helpcache</code> command.</p>
Procedure Cache Hit Ratio	<p>The percentage of time Sybase finds an available plan already in cache.</p> <p>If this is low, try to write more reusable code and/or consider increasing the size of the procedure cache.</p> <p>To determine the current size of the procedure cache, review the value of the <code>procedure cache size</code> parameter.</p>

## Disk

Metric	Description
DB Physical I/O Rate	The number of read and write operations that this Sybase server performed to or from disk every second. If this is high, drill in to the DPA Trends charts and review physical read and write wait types.
DB Physical Write Rate	The number of write operations this Sybase server performed to disk every second. If this is high, drill in to the DPA Trends charts and review queries with write wait types, such as "waiting for disk write to complete".

Metric	Description
DB Physical Read Rate	The number of read operations this Sybase server performed from disk every second. If this is high, drill in to the DPA Trends charts and review queries with physical read wait types, such as "waiting for i/o (read or write) to complete".
DB APF Read Rate	The number of Asynchronous Prefetch (APF) reads this Sybase server performed from disk every second. If this is high, drill in to DPA Trends charts and review queries with wait types of "waiting for an APF buffer read to complete".
Disk I/O Access Time	The average time to read from or write to disk.

## Network

Metric	Description
DB Network Send Rate	The number of bytes sent over the network every second for this database. If this is high, drill in to the DPA Trends charts and review queries with network wait types (for example, "waiting for network send to complete").
DB Network Receive Rate	The number of bytes received over the network every second for this database. If this is high, drill in to DPA Trends charts and review queries with network wait types (for example, "waiting for incoming network data").
DB Round-trip Time	The round-trip time when running "select 1" (includes network time but not connect time) on this database. If this is high, contact your network administrator to understand network latency.

## Sessions

Metric	Description
DB Active Sessions	The number of sessions actively performing work or waiting for a resource (excludes idle sessions).
DB Blocked Sessions	The number of sessions that are blocked because another session is using a needed resource.

## Waits


Metric	Description
Total Instance Wait Time	The total wait time for the instance.

## License Compliance

Metric	Description
Connected Users	The number of distinct users connected (that is, distinct logins) to this instance (even if the connection is idle). This value is typically used for per-user licensing.
Connected Devices	The number of distinct client machines connected to this instance (even if the connection is idle). This value is typically used for per-device licensing. It can also be used to approximate per-user licensing.
Sessions	The number of sessions connected to this instance (even if the connection is idle). This value is typically used for licensing based on number of concurrent connections.
Core Count	The number of cores used by the instance. This value is typically used for per-core licensing.

## Db2 metrics collected by DPA

The following sections list the metrics that DPA collects for Db2 self-managed database instances.

-  Learn how to [view these metrics](#) and [change the thresholds](#).
- For detailed information about resolving issues, click the Information link next to the metric on the Resources tab. The Information link is not available for all metrics.



## CPU

Metric	Description
O/S CPU Utilization	The percentage of CPU being used for the entire system (includes all databases on this machine). Potential solutions to a CPU bottleneck are to reduce the server load (tune those queries), get faster CPUs, or get more CPUs.

## Memory

Metric	Description
O/S Memory Utilization	The percentage of system memory being used for the entire system (includes all databases on this machine). If this is high, you might need to increase the amount of physical RAM in the server, reduce the load on the server, or change your server memory configuration.
Virtual Memory Utilization	The percentage of virtual memory being used.
DB Buffer Pool Hit Ratio	<p>The rate at which Db2 finds the data it needs in memory rather than having to read from disk. By itself, the buffer pool hit ratio is not very meaningful except for databases with undersized memory settings. Db2 might supply this value only if the DFT_MON_BUFFERPOOL monitoring switch is ON.</p> <p>Tuning queries and performing index optimization is the best way to increase buffer pool hit ratios.</p>
DB Package Cache Hit Ratio	The percentage of time when Db2 looks for an execution plan in the package cache and finds it. A low hit ratio indicates the <code>pckcachesz</code> parameter should be increased.
DB Catalog Cache Hit Ratio	The percentage of time when Db2 looks for an execution plan in the catalog cache and finds it. A low hit ratio indicates the <code>catalogcache_sz</code> parameter should be increased.

## Disk

Metric	Description
DB Physical Read Rate	<p>The number of reads performed from disk every second. If this is high, drill in to the DPA Trends charts and review queries waiting on physical read wait events.</p> <p> Db2 might supply this value only if the DFT_MON_BUFPOOL monitoring switch is ON.</p>
DB Physical Write Rate	<p>The number of writes performed to disk every second. If this is high, drill in to the DPA Trends charts and review queries waiting on write wait events.</p> <p> Db2 might supply this value only if the DFT_MON_BUFPOOL monitoring switch is ON.</p>



Metric	Description
DB Physical I/O Rate	The number of read and write operations performed to/from disk every second. If this is high, drill in to the DPA Trends charts and review queries waiting on physical read and write wait events.
	<b>i</b> Db2 might supply this value only if the DFT_MON_BUFPOOL monitoring switch is ON.

## Network

Metric	Description
DB Round-trip Time	The round-trip time when running "select 1 from sysibm.sysdummy1" against the database specified during registration (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

## Sessions

Metric	Description
DB Transaction Rate	The number of transactions being executed every second:  <code>commit_sql_stmts + int_commits + rollback_sql_stmts + int_rollbacks</code>
DB Connections Currently Executing	The number of sessions that are actively performing work or waiting for a resource (excludes idle sessions).
DB Blocked Sessions	The number of sessions that are waiting on lock waits because another session is using a needed resource.

## Waits

Metric	Description
Total Instance Wait Time	The total wait time for the instance.


## License Compliance

Metric	Description
Connected Users	The number of distinct users connected (even if the connection is idle). This value is typically used for per-user licensing.

Metric	Description
Connected Devices	The number of distinct client machines connected (even if the connection is idle). This value is typically used for per-device licensing. It can also be used to approximate per-user licensing.
Sessions	The number of sessions connected (even if the connection is idle). This value is typically used for licensing based on number of concurrent connections.
Core Count	The number of cores used by the instance. This value is typically used for per-core licensing.

## Azure SQL database metrics collected by DPA

The following sections list the metrics that DPA collects for Azure SQL database instances.

-  Learn how to [view these metrics](#) and [change the thresholds](#).
- For detailed information about resolving issues, click the Information link next to the metric on the Resources tab. The Information link is not available for all metrics.

### DTU

Metric	Description
DTU Consumption	The number of DTUs being used.
DTU Utilization	The percentage of the maximum DTUs being used. Use this value to determine the appropriate service tier for your needs.
DTU Limit	The DTU limit for this database instance.

### CPU

Metric	Description
CPU Utilization	The percentage of CPU being used based on the DTU limit.

### Memory

Metric	Description
Memory Usage Utilization	The percentage of memory being used.

Metric	Description
XTP Storage Utilization	The percentage of XTP storage utilization based on the DTU limit. This resource is available only for databases running on the Premium service tier. Zero percent is returned for the Basic and Standard service tiers.

## Disk

Metric	Description
Data I/O Utilization	The percentage of data I/O utilization based on the DTU limit.
Log Write Utilization	The percentage of log write Utilization based on the DTU limit
Database Storage Consumption	The percentage of the allotted storage space used by the database instance.
Database Size	The size of the data file in GB (rounded up to the nearest GB).

## Network

Metric	Description
Round-trip Time	The round-trip time when running "select 1" against this database (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

## Sessions

Metric	Description
Active Sessions	The number of sessions in this database actively performing work or waiting for a resource (excludes idle sessions).
Blocked Sessions	The number of sessions in this database that are blocked because another session is using a needed resource.
Max Worker Utilization	The percentage of Max Worker Utilization based on the database limit.
Max Session Utilization	The percentage of Max Session Utilization based on the database limit.

## Waits

Metric	Description
Total Instance Wait Time	Total wait time for the database instance.

## License Compliance

Metric	Description
Connected Users	The number of distinct users (that is, login names) connected to this database (even if the connection is idle).
Connected Machines	The number of distinct client machines connected to this database (even if the connection is idle). This value is typically used for per-device licensing. It can also be used to approximate per-user licensing.
Sessions	The number of sessions connected to this database (even if the connection is idle). This value is typically used for licensing based on number of concurrent connections.

## ASMI metrics collected by DPA

The following sections list the metrics that DPA collects for Azure SQL managed instances (ASMIs).

- Learn how to [view these metrics](#) and [change the thresholds](#).
- For detailed information about resolving issues, click the Information link next to the metric on the Resources tab. The Information link is not available for all metrics.

## CPU

Metric	Description
Signal Waits	<p>The percentage of overall time that sessions are waiting for a CPU to become available. Anything over 20% indicates a possible CPU resource bottleneck.</p> <p>Examine the overall wait events for the server as a whole. A high signal wait percentage could be due to an increased number of sessions, so examine the overall workload for the server as well. Take steps to either reduce the overall runtime for queries or reduce the total number of sessions.</p>
CPU Utilization	The amount of CPU being used as a percentage of the limit of the service tier.

## Memory

Metric	Description
XTP Storage Utilization	The percentage of available XTP Storage being used.
Page Life Expectancy	The number of seconds a page will stay in the buffer pool without references. A lower value (for example, under 300) indicates the buffer pool is under memory pressure and you should add more memory to the system (enable AWE on 32-bit systems) or find the process in Task Manager that is consuming outside of SQL Server.
O/S Memory Utilization	The percentage of memory being used for the entire system (includes all instances on this machine). If this is high and the Memory Paging Rate metric is high, you might need to increase the amount of physical RAM in the server, reduce the load on the server, or change the server memory configuration. Run <code>sp_configure</code> and review settings for "max server memory" and "min server memory" to determine amount of memory allocated to SQL Server.
Plan Cache Size	The current size of the SQL Server Plan Cache.
Buffer Cache Hit Ratio	The rate at which SQL Server finds the data blocks it needs in memory rather than having to read from disk for this instance. By itself, the buffer cache hit ratio is not very meaningful except for servers with undersized memory settings. Tuning queries and performing index optimization is the best way to increase buffer cache hit ratios. To see the current metrics for the buffer cache, run the following query:  <pre>select * from master..sysperfinfo where object_name like 'Buffer Manager'</pre>
Buffer Cache Size	The current size of the SQL Server Buffer Cache.
Procedure Cache Hit Ratio	The percentage of time when SQL Server looks for an execution plan in the procedure cache and finds it for this instance. If this is low, try to write more reusable code or consider increasing the size of the procedure cache. To see current metrics for the procedure cache, run the following query:  <pre>select * from master..sysperfinfo where object_name like '%Plan Cache%';</pre>
Log Bytes Flushed	The number of bytes of information being flushed per second.
Log Flushes	The number of log flushes that occur per second.

Metric	Description
SQL Compilations	The number of compilations performed by SQL Server per second. Compilations are a natural part of SQL Server operations but do utilize CPU and other resources. Compare this to the Batch Requests/sec metric to understand if this metric is too high. Minimizing compilations will help overall performance. For more information, see the following Microsoft Knowledgebase article: <a href="http://support.microsoft.com/kb/243588">http://support.microsoft.com/kb/243588</a> .
SQL Re-Compilations	The number of re-compilations performed by SQL Server per second. Re-compilations occur for many reasons but this number should typically be low.

## Disk

Metric	Description
Total I/O Wait Time	The sum of all I/O activity for all database files. If this is high: <ol style="list-style-type: none"> <li>1. Examine the current physical structure of databases on the server to see if it is possible to reduce I/O load by redistributing the database files to distinct disks.</li> <li>2. Examine queries and database design to determine if they can be tuned to reduce I/O.</li> </ol>
Total Read I/O Wait Time	The sum of all read I/O activity for all database files.
Total Write I/O Wait Time	The sum of all write I/O activity for all database files.
Data I/O Utilization	The average data I/O utilization as a percentage of the service tier limit.
Log Write Utilization	The average transaction log writes as percentage of the service tier limit.
O/S Disk Queue Length	The number of I/O operations waiting for disk drives to become available for the entire system (includes all instances on this machine). Spikes of high disk queue length may be normal, but if this is high for an extended period, you could have an I/O bottleneck. Drill in to DPA Trends charts to examine queries with I/O wait types during the timeframe.
Page Reads	The number of SQL Server physical reads from disk to memory. OLTP workloads are typically about 80-90 per second with higher values (or spikes) being an indication of insufficient storage performance, insufficient indexing, or not enough memory.

Metric	Description
Page Writes	The number of SQL Server physical writes from memory to disk. OLTP workloads are typically about 80-90 per second. If this is high (or spikes) it needs to be cross checked with lazy-writes/sec and checkpoints in order to determine if the issue might be due to low memory.
SQL Disk Read Latency	Disk read latency from dm_io_virtual_file_stats DMO.
SQL Disk Write Latency	Disk write latency from dm_io_virtual_file_stats DMO.

## Network

Metric	Description
Round-trip Time	The round-trip time when running "select 1" against this instance (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

## Sessions

Metric	Description
Transaction Rate	The number of transactions being executed every second in this instance (the Transactions/sec statistic from sysperfinfo for the instance).
Blocked Sessions	The number of sessions that are blocked in this instance because another session is using a needed resource.
Max Worker Utilization	Maximum concurrent workers (requests) as a percentage of the limit of the database's service tier.
Active Sessions	The number of sessions in this instance actively performing work or waiting for a resource (excludes idle sessions).
Max Session Utilization	Maximum concurrent sessions as a percentage of the limit of the database's service tier.
Batch Requests	The number of batches being executed by SQL Server every second.

## Waits

Metric	Description
Total Instance Wait Time	The total wait time for the database instance.

## License Compliance

Metric	Description
Sessions	The number of sessions connected to this instance (even if the connection is idle). This value is typically used for licensing based on number of concurrent connections.
Connected Users	The number of distinct users (that is, login names) connected to this instance (even if the connection is idle). This value is typically used for per-user licensing.
Core Count	The number of cores used by the instance. This value is typically used for per-core licensing.
Connected Devices	The number of distinct client machines connected to this instance (even if the connection is idle). This value is typically used for per-device licensing. It can also be used to approximate per-user licensing.

## PostgreSQL metrics collected by DPA

The following sections list the metrics that DPA collects for PostgreSQL databases.

 Learn how to [view these metrics](#) and [change the thresholds](#).

### Memory metric

Metric	Description
Buffer Cache Hit Ratio	The rate at which PostgreSQL finds the data blocks it needs in memory rather than having to read from disk.

### Disk metrics

Metric	Description
Blocks hit	The number of times disk blocks were found already in the buffer cache, so that a read was not necessary. This includes only hits in the PostgreSQL buffer cache, not the operating system's file system cache.
Blocks read	The number of disk blocks read in this database.



Metric	Description
Blocks Read Time	The average amount of block read I/O during the specified time interval. If the <code>track_io_timing</code> parameter is off, the value of this metric is always 0. For more information, see <a href="#">this KB article</a> .
Blocks Write Time	The average amount of block write I/O during the specified time interval. If the <code>track_io_timing</code> parameter is off, the value of this metric is always 0. For more information, see <a href="#">this KB article</a> .
Temp Files	The number of temporary files created by queries in this database. All temporary files are counted, regardless of why the temporary file was created (for example, sorting or hashing), and regardless of the <code>log_temp_files</code> setting.
Temp bytes written	The total amount of data in kilobytes written to temporary files by queries in this database. All temporary files are counted, regardless of why the temporary file was created, and regardless of the <code>log_temp_files</code> setting.
Write-ahead Log (WAL) Rate	The rate of the Write-ahead Log creation as a result of database transaction activity in MB per second.

## Network metric

Metric	Description
DB Round-trip Time	The round-trip time when running "select 1" (includes network time but not connect time) on this database.

## Sessions metrics

Metric	Description
Transaction Rate	The number of transactions being executed every second in this database instance.
Transaction Commit Rate	The number of transactions being committed every second in this database instance.
Transaction Rollbacks	The number of transactions in this database that have been rolled back.
All Sessions	All sessions, regardless of state.
Active Sessions	The number of transactions in the following state: The backend is executing a query.

Metric	Description
Blocked Sessions	The number of sessions in a blocked state.
Idle Sessions	The number of sessions in the following state: The backend is waiting for a new client command.
Idle in Transaction Sessions	The number of sessions in the following state: The backend is in a transaction, but is not currently executing a query.
Idle in Transaction (Aborted) Sessions	This state is similar to Idle in Transaction Sessions, except one of the statements in the transaction caused an error.
Fastpath Function Call Sessions	The number of sessions in which the backend is executing a fast-path function.
Other (State Monitoring Disabled) Sessions	This state is reported if <code>track_activities</code> is disabled in this backend.
Deadlocks	The number of deadlocks detected in this database instance.
Recovery Conflicts	The number of queries canceled due to conflicts with recovery in this database instance. Conflicts occur only on standby servers.

### Waits metric



Metric	Description
Total Instance Wait Time	The total wait time for the database instance.

### License Compliance metrics

Metric	Description
Connected Users	The number of distinct users (that is, login names) connected to this database instance, even if the connection is idle.
Connected Devices	The number of distinct client machines connected to this database instance, even if the connection is idle.

### Rows metrics

Metric	Description
Rows Operations	The number of rows inserted, updated, and deleted by queries in this database instance.

Metric	Description
Rows Inserted	The number of rows inserted by queries in this database instance.
Rows Updated	The number of rows updated by queries in this database instance.
Rows Deleted	The number of rows deleted by queries in this database instance.
Fetches vs. Rows Returned	Of the total number of rows that were scanned (Rows Returned), the percentage that contained data needed to execute the query (Rows Fetched). High values indicate that the database is executing queries efficiently. Low values indicate that the database is performing extra work because it is scanning a large number of rows that aren't required to process the query. For example, 10% means that the database is scanning 10 rows to use 1 row. Low values could indicate inefficient queries or missing indexes.
Rows Fetched	<p>The subset of scanned rows (Rows Returned) that contained data needed to execute the query. For example, take the following query:</p> <pre>SELECT * FROM customers WHERE country = 'Spain';</pre> <p>The <code>customers</code> table has 10,000 rows, and <code>country = 'Spain'</code> in 100 rows. The column is not indexed, and so a full table scan is required. The Rows Returned value is 10,000, but the Rows Fetched value is only 100.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> The Rows Fetched value is different than the number of rows returned to the client.</p> </div>
Rows Returned	<p>The total number of rows scanned by queries executed against this database instance.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> This value indicates rows returned by the storage layer to be scanned, <b>not</b> rows returned to the client.</p> </div>

## Vacuum metrics

Metric	Description
Transaction ID Space Taken	The percentage of space available to store Transaction IDs (XIDs) that is currently filled. this is the highest value across all databases in the database server.
Multixact ID Space Taken	The percentage of space available to store Multixact IDs (MXIDs) that is currently filled. this is the highest value across all databases in the database server.

Metric	Description
Autovacuum Worker Utilization	An indication of how busy the set of vacuum worker processes are. The <code>pg_stat_progress_vacuum</code> view provides information about current vacuuming processes. If this value is consistently high, consider increasing the value of the <code>autovacuum_max_workers</code> parameter.

## Checkpoint metrics

Metric	Description
Requested Checkpoints Ratio	The ratio of requested checkpoints to total checkpoints (requested and scheduled). The percentage should be low—optimally 0%.
Requested Checkpoints	The number of unscheduled checkpoints requested by client statements because the WAL size has reached its threshold ( <code>max_wal_size</code> ). Requested checkpoints can cause client backend waits. Consider reconfiguration of checkpoint related settings ( <code>checkpoint_timeout</code> , <code>checkpoint_completion_target</code> , and <code>max_wal_size</code> ).
Scheduled Checkpoints	The number of scheduled checkpoints processed in the background, without affecting client statements. Scheduled checkpoints should not cause client backend waits.

## Replication metrics

Metric	Description
Replication Lag	The replication lag between the primary database and all replica databases. An increase in the replication lag indicates a growing number of transactions that are not yet replicated and at risk of not being replicated if the primary database fails.


## Cache Eviction metrics


Metric	Description
Dirty Buffers Evicted by Client Backends	The number of times client backends were delayed by being forced to write and free (evict) buffers themselves, instead of the buffers being evicted asynchronously by the background writer.
Dirty Buffers Evicted by Client Backends Ratio	The ratio of buffers written and freed (evicted) by a client backend to the total number of evictions.

Metric	Description
Dirty Buffers Evicted by Background Writer	The number of buffers written and freed (evicted) due to the PostgreSQL background writer.
Dirty Buffers Evicted by Background Writer Ratio	The ratio of buffers written and freed (evicted) due to the PostgreSQL background writer to the total number of evictions.
Dirty Buffers Evicted by Checkpoints	The number of buffers written and freed (evicted) due to a checkpoint execution. Higher values indicate an increased need for checkpoints.
Dirty Buffers Evicted by Checkpoints Ratio	The ratio of buffers written and freed (evicted) due to a checkpoint execution to the total number of evictions. Higher values indicate an increased need for checkpoints.
Total Dirty Buffers Evicted	The total number of dirty buffers evicted by checkpoints, background writer, and client backends.

## VM metrics collected by DPA

The following sections list the metrics that DPA collects from virtual machines (VMs). For database instances that run on a VM, these metrics are displayed in addition to the metrics collected for the database type.

 DPA collects VM metrics only if you [register the VM for monitoring](#). Monitoring a VM requires a [VM license](#).

-  Learn how to [view these metrics](#) and [change the thresholds](#).
- For detailed information about resolving issues, click the Information link next to the metric on the Resources tab. The Information link is not available for all metrics.

## CPU

Metric	Description
VM CPU Usage Percentage	Actively used CPU as a percentage of the total available virtual CPU in the virtual machine. Note that this is the host's view of the CPU usage, not the guest O/S view, so the values may differ.

If this value is high, check the VM CPU Ready Time:

- If VM CPU Ready Time is also high, the host has under-allocated CPU resources to the VM. (See VM CPU Ready Time below.)
- If VM CPU Ready Time is not high and you are not experiencing a performance problem, high CPU usage values are not a cause for concern.
- If VM CPU Ready Time is not high but you are experiencing a performance problem, you can address the issue in either of the following ways:

- Increase the CPU resources provided to the VM.

To do this, you can add vCPUs to the VM, migrate the VM to a host with more powerful processors, or add additional VMs running the same application and then balance the workload.

- Increase the efficiency with which the VM uses CPU resources.

To do this, you can tune the queries with long Memory/CPU wait types or tune the non-database applications using the most CPU.

---

Metric	Description
VM CPU Ready Time	<p>The percentage of time that the virtual machine was ready to use CPU resources, but could not get scheduled to run on the physical CPU. This value is the average across all CPUs.</p> <p>If this metric is high, check the Host CPU Usage:</p> <ul style="list-style-type: none"><li>• If Host CPU Usage is normal, the VM could have under-allocated CPU resources. If the VM has been configured with a CPU limit, consider raising or removing the limit. Or use resource controls to give higher priority to this VM, which will allocate more CPU resources to it.</li><li>• If Host CPU Usage is high, this could indicate a host over-commitment of CPU resources. Consider:<ul style="list-style-type: none"><li>◦ Reducing the number of VMs running on the host.</li><li>◦ Increasing the available CPU resources by adding the host to a DRS cluster.</li><li>◦ Increasing the efficiency with which VMs use CPU resources by tuning SQL statements and non-database applications.</li><li>◦ Using resource controls to direct available resources to critical VMs.</li></ul></li></ul>
Host CPU Usage	<p>Actively used CPU as a percentage of the total available CPU on the machine. If this metric is high, determine if the VM CPU Ready Time is also high.</p>
VM CPU Usage MHz	<p>The average amount of CPU (in MHz) actively used by the VM (for all vCPUs configured for the VM). This is the host's view of the CPU usage, not the guest operating system view, so the values may differ. Typically the host view of CPU is more accurate.</p>
VM Total CPU Usage Time	<p>The total amount of time (in milliseconds) that the VM spent using the virtual CPUs (that is, the sum of time spent on each virtual CPU during the time period).</p>

Metric	Description
VM Total Co-Stop Time	<p>The amount of wait time incurred because the VM in which the database is running has to wait on physical CPU resources allocated to other VMs.</p> <p>If the database instance is on a VM configured to use multiple vCPUs, co-stop delays can cause long Memory/CPU wait times. Co-stop delays occur when a VM is not being scheduled to run consistently because it has to wait on vCPU resources to be freed from other VMs contending for those vCPUs.</p> <p>If this value is not near 0, consider taking one of the following actions to reduce co-stop delays:</p> <ul style="list-style-type: none"> <li>• Decrease the number of vCPUs on the VM.</li> <li>• Add additional CPUs to the pool available to the VMs.</li> <li>• Use vMotion to migrate other VMs to a different host to reduce contention.</li> </ul>
VM Co-Stop	The percentage of time the VM has been waiting on physical CPU resources allocated to other VMs. If this value is above 3%, consider the actions listed above to reduce co-stop delays.

## Memory

Metric	Description
VM Active Memory Usage	<p>Memory that is actively in use (that is, used currently or in the recent past) as a percentage of virtual machine configured memory.</p> <p>While a VM may have been allocated large amounts of memory, it is possible that the OS and applications are only using a small percentage of what the VM was assigned. Inactive memory is subject to being "ballooned" (reclaimed by other VMs) when memory is scarce.</p> <p>When the active memory for all VMs exceeds the total host memory, it indicates host memory saturation. As a result, host-level memory swapping typically occurs.</p>
VM Memory Granted	Memory that has been given to the virtual machine by the host, not including overhead. Typically, VMs are granted increasing amounts of memory over time until reaching the configured VM memory size.



Metric	Description
VM Memory Swap In Rate	<p>The rate at which memory is swapped in from disk. A value greater than 0 indicates that performance is suffering due to lack of memory. This is typically caused by memory being previously swapped out, memory over-commitment (many virtual machines with high amounts of active memory), or a problem with the balloon driver. Consider the following possible solutions:</p> <ul style="list-style-type: none"><li>• Reduce the level of memory over-commit.</li><li>• Enable the balloon driver in all VMs.</li><li>• Reduce memory reservations.</li><li>• Use resource controls to dedicate memory to critical VMs.</li></ul>
VM Memory Swap Out Rate	<p>The rate at which memory is swapped out to disk. High values indicate a problem with lack of memory that is causing performance to suffer. This is typically caused by either memory over-commitment (many virtual machines with high amounts of active memory) or a problem with the balloon driver.</p>
Host Memory Usage	<p>The actual memory usage on the host (total consumed memory / total machine memory). High host memory usage is not necessarily a problem, but could indicate host memory over-commitment (or looming over-commitment). Check to see if memory swapping is occurring by looking at memory swap in/out rates, which is a clear indicator of host memory over-commitment.</p>
VM Memory Balloon Size	<p>The amount of virtual machine memory that is currently claimed by the balloon driver. If high amounts of ballooning are occurring, check for high Memory Swap In/Out Rates which would indicate performance problems.</p>
VM Memory Balloon	<p>The percentage of the virtual machine memory that is currently claimed by the balloon driver. This is not necessarily a performance problem, but shows the host starting to take memory from VMs that need less memory and assigning it to VMs with large amounts of active memory. If high amounts of ballooning are occurring, check for high Memory Swap In/Out Rates which would indicate performance problems.</p>
VM Memory Overhead	<p>The amount of memory used to run the virtual machine. Configuring a virtual machine with excess memory or excess virtual CPUs will unnecessarily increase the overhead.</p>

## Disk

Metric	Description
VM Disk Commands	The number of disk commands issued by the virtual machine. High disk usage could be due to guest swapping, which you can investigate using OS analysis tools. VMs configured with insufficient memory can also cause excessive guest swapping and, in turn, high disk usage.
VM Disk Commands Aborted	<p>The number of disk commands that were aborted. This typically occurs when storage demand is excessively high, or when storage is not properly configured to handle the I/O load.</p> <p>Beyond re-balancing load, there is typically little that can be done from within vSphere to solve problems related to slow or overloaded storage. Follow the guidelines from your storage vendor to monitor the demand being placed on the storage device, and follow the vendor-specific configuration recommendations to configure the device for the demand. If the device is not capable of satisfying the I/O demand with good performance, distribute the load among multiple devices, or obtain faster storage.</p>
VM Disk Bus Resets	<p>The number of disk bus reset commands issued. This typically occurs when storage demand is excessively high, or when storage is not properly configured to handle the I/O load.</p> <p>Bus Resets occur when the disk subsystem times out and commands are canceled and retried. This happens when the HBA device is overloaded or its queue depth is exhausted.</p>
VM Disk Read Rate	<p>The average rate at which data is read from each virtual disk on the virtual machine.</p> $\text{read rate} = \# \text{ blocks read per second} \times \text{block size}$
VM Disk Write Rate	<p>The average rate at which data is written to each virtual disk on the virtual machine.</p> $\text{write rate} = \# \text{ blocks read per second} \times \text{block size}$
VM Disk Usage Rate	The average disk I/O rate across all virtual disks on the virtual machine.
Host Max Total Disk Latency	The highest latency value across all disks used by the host. Latency measures the time taken to process a disk command issued by the guest OS to the virtual machine. High latency is a key indicator of slow storage.

Metric	Description
Host Disk Read Rate	<p>The average rate at which data is read from each LUN on the host.</p> $\text{read rate} = \# \text{ blocks read per second} \times \text{block size}$ <p>If your database instance is suffering from disk I/O performance related issues, it's possible that another VM on the same host is consuming high amounts of disk resources and causing delays for this VM. To understand that relationship, check the Physical I/O rate from the database instance compared to this metric. If this metric is much higher than the database metric, another VM might be causing the issue. If not, this VM might be putting too many demands on the underlying disk devices.</p>
Host Disk Write Rate	<p>The average rate at which data is written to each LUN on the host.</p> $\text{write rate} = \# \text{ blocks written per second} \times \text{block size}$

## Disk Device

Metric	Description
Host Disk Device Read Rate	<p>The average rate at which data is read from a specific LUN on the host (across all VMs on the host).</p>
Host Disk Device Write Rate	<p>The average rate at which data is written to a specific LUN on the host (across all VMs on the host).</p>
Host Disk Device Read Latency	<p>The average time taken to process a SCSI read command issued from the Guest OS to the virtual machine (across all VMs).</p> <p>Expected disk latencies depend on the nature of the storage workload (for example, read/write mix, randomness, and I/O size) and the capabilities of the storage subsystems.</p>
Host Disk Device Write Latency	<p>The average time taken to process a SCSI write command issued from the Guest OS to the virtual machine (across all VMs).</p>

## Network

Metric	Description
VM Data Receive Rate	<p>The average rate at which data is received on the virtual machine. This represents the receive bandwidth of the network.</p>

Metric	Description
VM Data Transmit Rate	The average rate at which data is transmitted on the virtual machine. This represents the transmit bandwidth of the network.
VM Network Packets Received	The number of packets received across all vNICs on the virtual machine.
VM Network Packets Transmitted	The number of packets transmitted across all vNICs on the virtual machine.
Host Dropped Received Packets	The number of dropped received packets across all physical NICs on the host.
Host Dropped Transmitted Packets	<p>The number of dropped transmitted packets across all physical NICs on the host. The following problems can cause the guest OS to fail to retrieve packets quickly enough from the virtual NIC:</p> <ul style="list-style-type: none"><li>• High CPU utilization</li><li>• Guest OS driver configuration</li></ul> <p>Solutions are all related to ways of improving the ability of the guest OS to quickly retrieve packets from the virtual NIC. You can:</p> <ul style="list-style-type: none"><li>• Increase the CPU resources provided to the VM.</li><li>• Increase the efficiency with which the VM uses CPU resources.</li><li>• Tune network stack in the Guest OS.</li><li>• Add additional virtual NICs to the VM and spread network load across them.</li></ul>

# DPA user accounts

Use the following topics to create user accounts, assign privileges, and specify how users will log in to DPA:


- [DPA roles and privileges](#)
- [Create a user account](#)
- [User authentication options](#)
- [Configure Active Directory or LDAP](#)

## DPA roles and privileges

When you [add user accounts in DPA](#), you assign each user a role. The role determines the user's privileges.

### Administrator role

Administrators have access to all DPA functionality, including all setup, administration, and support options.

 DPA requires at least one Administrator account, which is created during installation.

Only administrators can perform certain actions, such as:

- Register and unregister database instances and VMs
- Allocate licenses
- Run advanced support utilities
- Edit system-wide Advanced Options
- Start and stop all monitors
- Create, edit, and delete report schedules
- Create, edit, and delete alert groups
- Create, edit, and delete email templates for alert notifications
- Configure the mail server
- Administer users, contacts, and contact groups

## Read Only on All Instances role

Users with this role can perform the following actions for **all** database instances:

- View performance data and metrics
- Run reports and view existing report groups
- View existing alerts
- View logs


## Custom Privileges role

The Custom Privileges role specifies which privileges a user has, and which database instances these privileges apply to. Use this role to:

- Prevent users from seeing data about certain database instances
- Give users privileges to manage monitoring options, alerts, and reports without granting them full administrative privileges

When you assign this role to a user, you can grant any of the following privileges. Privileges can apply to all database instances or only selected instances.

Privilege	Actions allowed against selected database instances
View Data	<ul style="list-style-type: none"> <li>• View performance data and metrics</li> <li>• Run reports and view existing report groups</li> <li>• View logs</li> </ul>
Manage Reports	Create, edit, and delete report groups
View Alerts	View existing alerts
Manage Alerts	<ul style="list-style-type: none"> <li>• Create, edit, and delete alerts</li> <li>• View existing alert groups</li> </ul>
Manage Monitoring	<ul style="list-style-type: none"> <li>• Create, edit, and delete blackout periods for monitoring</li> <li>• Manage I/O configuration</li> <li>• Update Advanced Options for a specific database instance</li> <li>• Add annotations</li> <li>• Exclude SQL statements from trend charts</li> <li>• Start and stop individual monitors</li> </ul>

 Users with Manage Monitoring permissions cannot see the charts at the top of the DPA homepage.

## Create a DPA user account and assign privileges

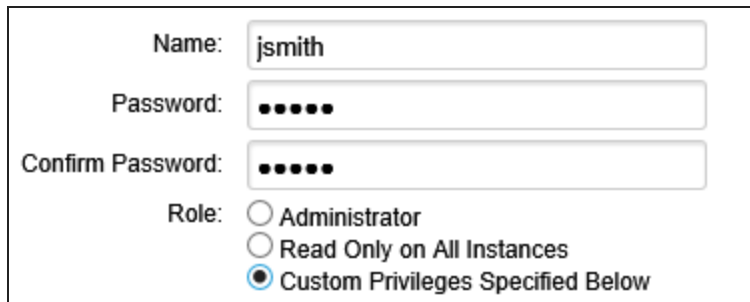
You must add a user account for each person who needs to log in to DPA. Each user is assigned a role, which determines the user's permissions.

**i** Optionally, you can [integrate DPA with your company's Active Directory \(AD\) or LDAP service](#). If you do this:

- Users can log in to DPA with their domain accounts.
- You can add AD or LDAP groups to DPA and assign privileges to each group.

Before you add users, determine who needs access to DPA and which privileges each user needs. For more information about the available options, see [DPA roles and privileges](#).

1. On the DPA menu, click Options.
2. Under Administration > Users & Contacts, click User Administration.
3. On the User Administration page, click Create User.
4. Enter a unique user name and a password.
5. Specify the user's privileges:
  - To assign privileges associated with [predefined roles](#), select Administrator or Read Only on All Instances.
  - To assign custom privileges:
    - a. Select Custom Privileges Specified Below.



Name:

Password:

Confirm Password:

Role:  Administrator  
 Read Only on All Instances  
 Custom Privileges Specified Below

- b. To grant access to data from all database instances, select [privileges](#) in the top row. To limit access, select privileges for each database instance.

**i** The View Data privilege is automatically selected when you select any higher privilege.

In the example below, the user can access data from only one database instance. This user can make changes to monitoring options, run reports, and view existing alerts for the selected instance.

Database Instance	Type	View Data	Manage Reports	Alerts	Manage Monitoring
Change All →		<input type="checkbox"/>	<input type="checkbox"/>	None ▼	<input type="checkbox"/>
DPA-SUSE-MYSQL56:3306	MySQL	<input type="checkbox"/>	<input type="checkbox"/>	None ▼	<input type="checkbox"/>
DPA-WIN-MYSQL57:3306	MySQL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	View ▼	<input checked="" type="checkbox"/>

6. Click Save.

If you configured DPA to point to your Active Directory or LDAP server, you will see an option to either create a user or a group. The group corresponds to a group in Active Directory or LDAP.

## DPA user authentication options

DPA supports Active Directory (AD) and Lightweight Directory Access Protocol (LDAP) authentication. Using your existing authentication infrastructure eliminates the need to duplicate your user accounts in DPA. After you configure AD or LDAP authentication, users can log in with their domain account or a custom user account created by DPA.

### AD user authentication

DPA integrates with Windows Active Directory (AD). DPA uses the security group information from AD to assign permissions to groups. To configure DPA user authentication and permissions using AD, see [Configure Active Directory or LDAP](#).

If your repository database is Azure SQL and you are monitoring one or more Azure SQL databases, you can use Azure AD authentication in DPA. To configure DPA user authentication and permissions using Azure AD, see [Use Azure AD authentication in DPA](#).

### LDAP user authentication

DPA integrates with most LDAP implementations to assign permissions to groups. To configure DPA user authentication and permissions using LDAP, see [Configure Active Directory or LDAP](#).

### Single sign-on

Using single sign-on (SSO), your AD users can log in to DPA without re-entering the domain credentials they used to log in to their operating system. [Before you configure DPA for SSO](#), configure DPA for AD authentication.



## Common Access Cards

You can use a Common Access Card (CAC) to log in to Windows and DPA. Before using a CAC, configure DPA for AD, and then for SSO as described in the sections above.

## Configure DPA to use Active Directory or LDAP

To use AD or LDAP user authentication in DPA:

1. Gather the following information from your domain administrator:
  - Directory service type: AD or LDAP
  - Domain name
  - Port number: Used to connect to the directory service
  - User: The domain user DPA uses to query the directory for users and groups
  - Password: The password of the domain user, preferably one that does not expire
2. In DPA, click Options. Then, under Administration > Users & Contacts, click Configure AD/LDAP.
3. Select the type of directory service you have: Active Directory or LDAP.
4. Click Next.

## Connection information

### Domain name

Enter the domain name.

 SolarWinds recommends using a domain name, not the name of a specific domain controller.

Do you have multiple domains?

If your domain users authenticate from a different domain other than the domain specified here, you must connect to the global catalog ports 3268 or 3269. The domain users must belong to a universal group, and that universal group must be added under Options > Administration > Users & Contacts > User Administration.

### Port

Select the port number.

If you use a unique port, select Other non-standard port. Enter the port number, and select SSL if required.


## User and Password

DPA uses this user to search the directory service for users and groups.

Active Directory user name

For the AD user name, use one of the following formats:

- Distinguished Name (DN): `cn=BobSmith, cn=Users, dc=domain, dc=local`
- User Principal Name (UPN): `bsmith@domain.local`

 See [this article](#) for information about valid characters for Active Directory user names.

LDAP user name

For the LDAP user name, use the following format:

- Distinguished Name (DN): `cn=BobSmith, cn=Users, dc=domain, dc=local`

## Did the connection test fail?

If you use an SSL port and the verification fails, DPA must import its certificate. Click Yes on the confirmation window to try again.

## Base search location

### Base DN

Use the default

SolarWinds recommends selecting the default, so DPA uses the detected base DN from the previous step.

Example of default base DN: `dc=east, dc=acme, dc=com`

Use a custom value

You may use a value other than the default base DN. For example: You use a global catalog that supports multiple domains, and you want to broaden the scope of the search.

Example for multiple domains: `dc=acme, dc=com`

## Advanced settings

If this is your first time using this wizard, do not use the advanced settings.

Only use advanced settings if you completed this wizard and you experience slow domain user logins or group searches.

Are domain user logins slow?

Set the User Search Base value if domain user logins take a long time.

If your company has one domain, specify the location in the directory tree that contains all of the domain users that will use DPA.

If you do not know what to put here, ask the domain administrator of your company the following questions:

"What folder, or organization unit (OU), in the directory tree of the domain contains all of the users? I must specify a search base for users. What is the distinguished name of the folder?"

Example: `cn=users OR ou=users`

Are domain group searches slow?

Set the Group Search Base value if domain group searches in User Administration take a long time.

Specify the location in the directory tree that contains all of the groups to which DPA users belong.

If your company has multiple domains, you can enter the group search bases individually. After you add groups to DPA using the group search base from one domain, update this wizard to specify a group search base in another domain.

If you do not know what to put here, ask your the domain administrator of your company the following:

"What folder, or organization unit (OU), in the directory tree of the domain contains all of the groups? I must specify a search base for groups. What is the distinguished name of the folder?"

Example: `cn=groups OR ou=groups`

## Summary

Confirm the information for configuring DPA with your directory service, and click Finish.


 You must restart the DPA server for the settings to take effect.

## Configure authentication and permissions for groups of users

After you have set up DPA to use Active Directory or LDAP, do the following:

1. In AD or LDAP, determine which groups contain the users that you want to grant access to DPA. You may need to create a group if a suitable group does not exist.
2. On the DPA menu, click Options.
3. Under Administration > Users & Contacts, click User Administration.

4. Click Add Active Directory Group or Add LDAP Group.
5. Click Search for a Group.
6. Find and select the group you want and click Save.
7. Assign privileges to the group, just as you would for a user. This assigns those permissions to the domain users who are members of the group.

 DPA does not support single sign-on (SSO) for individual accounts. It only supports AD or LDAP groups.

8. Click Save.

All domain users in the selected group can log in to DPA using their domain credentials. The users have the privileges you set up for the group in DPA.

You can add multiple AD or LDAP groups in DPA. If a domain user is a member of more than one group, DPA grants them the combined privileges from all of their groups.

## Log in to DPA

When you enter the domain user name and password in the DPA login screen, DPA searches your directory service for a matching user name, and then authenticates using the password. If the domain user belongs to one of the groups that you configured as a DPA custom user, the login succeeds.

### Name formats for AD login

DPA supports three types of login name formats for Active Directory:

- **SAM account name:** `username`
- **User principal name:** `username@domain.local`
- **NT/AD:** `domain\username`

### User name for LDAP

The user name used by DPA is the LDAP user object `uid` attribute.

## DPA alerts

Use DPA alerts to become aware of issues and address them proactively before they affect end users. Set thresholds on key wait time statistics, resource metrics, or standard administration indicators. The result is improved customer service, fewer help desk tickets, and increased compliance with database service-level agreements.

DPA provides the following types of alerts:

- [Wait Time alerts](#) are triggered when wait time exceeds a user-defined threshold, or when wait time is much higher than expected (an [anomaly](#)).
- [Resources alerts](#) are triggered when a resource metric, such as CPU utilization or memory usage, exceeds its threshold.
- [Administrative alerts](#) are used to monitor the health of the database system.
- [Custom alerts](#) are user-specified queries that are run against the monitored database or the DPA repository.

To work with DPA alerts, see the following topics:









- [View the status and history of DPA alerts](#)
- [Configure a DPA Wait Time alert](#)
- [Configure a DPA Resources alert](#)
- [Configure a DPA Administrative alert](#)
- [Configure a DPA Custom alert](#)
- [Send SNMP traps from DPA alerts](#)
- [Stop DPA alerts for a period of time](#)
- [Create a DPA alert group](#)
- [Create and manage DPA contacts and contact groups](#)
- [Notification policy for DPA alerts](#)

## View the status and history of DPA alerts

View current and previous statuses of DPA alerts. You can select alert history records to purge from the logs.


## 1. On the DPA menu, click Alerts.

The Alert Status tab shows information about the most recent evaluation of each DPA alert against each database instance. It includes the alert status (for example, High, Normal, or Inactive), the current value, and the last time the status changed.

Alert Status							All Database Instances	
Status	Name	Database Instance	Type	Current Value	Last Change			
 NORMAL	Total SQL Wait Time for a Single SQL	DPA-MANAGED	Total SQL Wait Time for a Single SQL	0 seconds	21 May 2019 06:41	<a href="#">Detail</a>	<a href="#">History</a>	
 NORMAL	Total SQL Wait Time for a Single SQL	DPABIRDYS	Total SQL Wait Time for a Single SQL	0 seconds	10 May 2019 04:28	<a href="#">Detail</a>	<a href="#">History</a>	
 NORMAL	Total SQL Wait Time for a Single SQL	DPADB2111.50000	Total SQL Wait Time for a Single SQL	0 seconds	10 May 2019 04:28	<a href="#">Detail</a>	<a href="#">History</a>	
 NORMAL	Total SQL Wait Time for a Single SQL	CUBEM.IGNITE.LOCAL:3306	Total SQL Wait Time for a Single SQL	0 seconds	10 May 2019 04:28	<a href="#">Detail</a>	<a href="#">History</a>	
 NORMAL	Total SQL Wait Time for a Single SQL	DPAORA12_DPAORA12	Total SQL Wait Time for a Single SQL	0 seconds	10 May 2019 04:28	<a href="#">Detail</a>	<a href="#">History</a>	
 HIGH	Total Database Instance Wait Time alert	JOHNLENNON via MUSICDB-LIS	Total Database Instance Wait Time	9 seconds	29 Jul 2019 11:46	<a href="#">Detail</a>	<a href="#">History</a>	
 MEDIUM	Total Database Instance Wait Time alert	DPASQL2K17	Total Database Instance Wait Time	5 seconds	29 Jul 2019 13:14	<a href="#">Detail</a>	<a href="#">History</a>	
 NORMAL	Total Database Instance Wait Time alert	DPAORA12_DPAORA12	Total Database Instance Wait Time	0 seconds	29 Jul 2019 09:14	<a href="#">Detail</a>	<a href="#">History</a>	

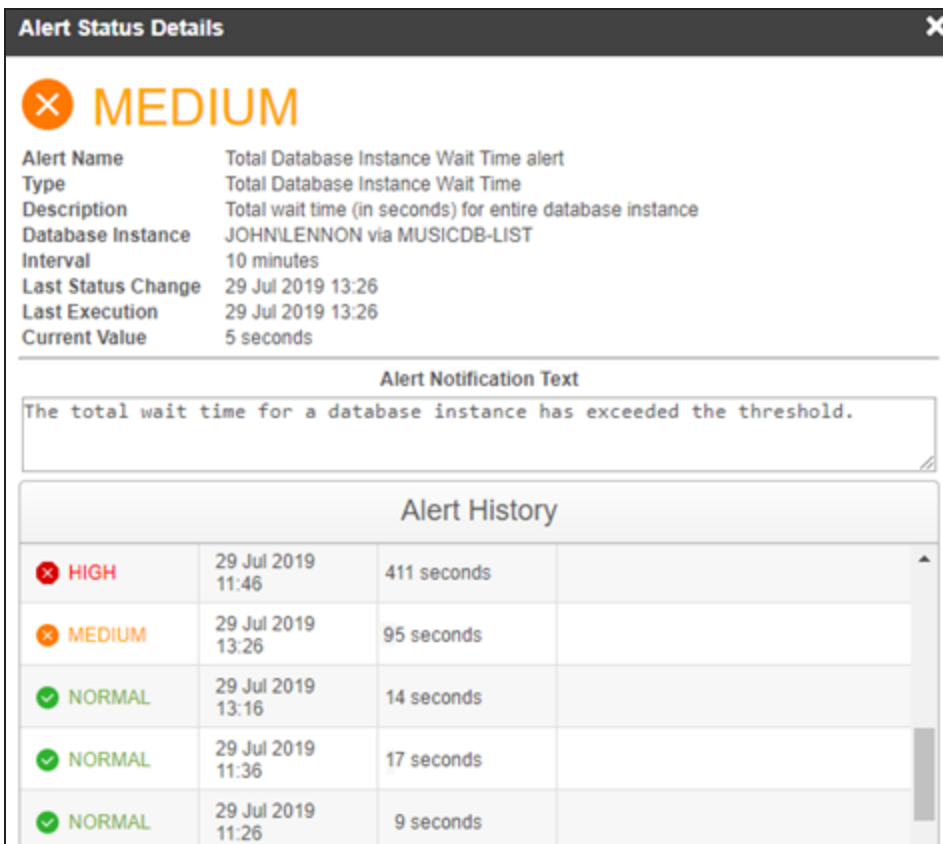
## 2. Scan, sort, or filter to find the alerts you are interested in:

- To filter the list of alerts, choose a database instance from the drop-down menu in the upper-right corner.
- Click a column heading to sort the list of alerts by the value of that column.

 Click the name of the alert to open it for editing. See [Configure a DPA Wait Time alert](#) for information about the available options.

## Display alert details

Click the Details button to display additional information about an alert's definition, last execution time, current value, and history.



**Alert Status Details**






**MEDIUM**

Alert Name: Total Database Instance Wait Time alert  
Type: Total Database Instance Wait Time  
Description: Total wait time (in seconds) for entire database instance  
Database Instance: JOHNLENNON via MUSICDB-LIST  
Interval: 10 minutes  
Last Status Change: 29 Jul 2019 13:26  
Last Execution: 29 Jul 2019 13:26  
Current Value: 5 seconds

**Alert Notification Text**

The total wait time for a database instance has exceeded the threshold.

**Alert History**

 HIGH	29 Jul 2019 11:46	411 seconds	
 MEDIUM	29 Jul 2019 13:26	95 seconds	
 NORMAL	29 Jul 2019 13:16	14 seconds	
 NORMAL	29 Jul 2019 11:36	17 seconds	
 NORMAL	29 Jul 2019 11:26	9 seconds	

## View or purge alert history

Click the History button to display up to 5,000 lines of alert history. By default, the Alert History page lists the results of evaluating the alert against the database on the associated line of the Alert Details tab.

To change the default display:

1. Select the desired values for Status, Date Range, Number of Records, and Database Instances.
2. Click Apply Filter.

To purge alert history records:

1. Use the Status, Date Range, Number of Records, and Database Instances fields to identify the records you want to purge.

To purge all history before a certain date, enter that date as the End Date and leave the Start Date blank.

2. (Optional.) Click Apply Filter to verify that only the records you want to purge are selected.
3. Click Purge Log, and then click Yes at the confirmation prompt.


All records meeting your criteria are removed.

## Configure a DPA Wait Time alert

Wait Time alerts notify you when the amount of time users or applications waited on the database was high. These alerts are triggered when wait time exceeds a user-defined threshold, or when wait time is much higher than expected (an [anomaly](#)).

 For information about other DPA alert categories, see [DPA alerts](#).


1. From the DPA main menu, click Alerts.
2. Click the Manage Alerts tab.
3. Do one of the following:
  - To create a new alert, select Wait Time as the Alert Category, select the Alert Type, and then click Create Alert.

 To find out more about each alert type, select it to display a description on the right.

- To edit an existing alert, click the alert name.
4. In the Alert Information section:
    - a. Enter a unique name.
    - b. If you want to disable the alert, clear the Active check box.
    - c. Select the execution interval.

The execution interval specifies how often the alert runs and the amount of data that DPA examines. For example, if the execution interval is 10 minutes, DPA executes the alert every 10 minutes and examines the last 10 minutes of data to determine whether to trigger the alert. DPA recommends an execution interval of **at least 10 minutes** to prevent unnecessary alerts from a single slow execution or temporary condition.

- d. Enter the notification text to be sent with the email notification. Include an explanation of the issue and the suggested resolution.

 If you apply a custom [email template](#) to this alert, the email notification includes the notification text if the email template contains one of the following variables:

- Alert Notification text: [=alert.notificationText]



**i** • Results: [=dpa.body] (included by default)

### Alert Information

Type	Average Wait Time for a Single SQL
Description	Average execution time (in seconds) for the specified SQL
Alert Name	<input type="text" value="Average Wait Time for SELECT FROM CUST OUTER JOIN"/>
Active	<input checked="" type="checkbox"/>
Execution Interval	<input type="text" value="10"/> <input type="text" value="Minutes"/>

Notification Text - *Explanation or resolution steps to be sent with alert email*

The average wait time for SELECT FROM CUST OUTER JOIN is above 30 seconds for the execution interval. Kill this query to prevent performance problems.

5. Select the database instances that the alert applies to.
6. If the alert type requires parameters, under Alert Parameters:
  - a. Click Search.
  - b. If necessary, select a database instance at the top of the Search dialog box.
  - c. Enter a search string (for example, part of the SQL statement name or wait type).

**i** By default, search results for SQL statements are sorted by wait time, with the highest waits first. To list them in alphanumeric order by name or hash value, clear the Order results by wait time check box, and then click Search again.





- d. Select a value and click OK.
7. For all Wait Time alerts **except** the Database Instance Wait Time Anomaly alert, specify the thresholds for each alert level you want to enable.

**i** [Alert thresholds for anomalies](#) have default values that can be changed through advanced configuration options.

- Leave the Max value for the highest level **blank** to alert on anything above the minimum value for that level.
- If you configure multiple levels, the Max value for lower levels must **equal** the Min value


for the next higher level.

- When you enter a Max value for a level, DPA alerts at that level when the value is **greater than or equal to** the Min value but **less than** the Max level. For example, if the Min value is 5 and the Max value is 10, DPA will alert at that level when the value is 5 or when the value is 9.99, but **not** when the value is 10.


	Min (occurrences)	Max (occurrences)	
 HIGH	<input type="text" value="10"/>	<input type="text"/>	<div style="border: 1px solid black; padding: 5px;">           A high-level alert is triggered when the value is 10 or greater.         </div>
 MEDIUM	<input type="text" value="5"/>	<input type="text" value="10"/>	
 LOW	<input type="text"/>	<input type="text"/>	<div style="border: 1px solid black; padding: 5px;">           A medium-level alert is triggered when the value is 5 to anything <b>less than</b> 10.         </div>
 INFO	<input type="text"/>	<input type="text"/>	

Select the person or group who gets notified when each alert level is triggered and when the alert is broken. (The alert status is set to Broken if an error occurs during execution.)

To send notifications when the alert returns to Normal, select a recipient for Normal.

-  • If you have not added the person or group as a contact in DPA, click Add Contact or Add Contact Group and [create the contact or group](#).
- Select an [SNMP contact](#) to send SNMP traps when the alert is triggered. Select an SNMP contact for Normal to send a clearing notification when the alert status returns to Normal.


8. Verify or change the [notification policy](#).

-  To send notifications when the alert returns to Normal, the notification policy must be `Notify when level changes`.

9. Select the [email template](#) that defines the contents of the email notifications sent by this alert.

10. Click Email Preview to see an example of the email that will be generated using the selected email template and contact information.

If the alert applies to multiple database instances, select an instance in the Email Preview dialog box and click OK. After reviewing the email, you can select a different database instance or click Cancel to close the Email Preview dialog box.

-  The email sent to users might not exactly match the preview because some alert parameters cannot be evaluated during a preview.

11. Click Test Alert to test the alert and view the current alert level. The test does not generate an email.
12. Click Save.

## Configure a DPA Resources alert

Resources alerts are triggered when a resource metric, such as CPU utilization or memory usage, exceeds its threshold. A Resources alert can monitor a single resource metric (such as Buffer Cache Hit Ratio) or all metrics in a resource category (such as Memory).


 For information about other DPA alert categories, see [DPA alerts](#).

## Verify resource thresholds

Thresholds for Resources alerts are specified in the resource settings, not within the alert. You can [view or change](#) them if needed.

## Create or edit a Resources alert


1. From the DPA main menu, click Alerts.
2. Click the Manage Alerts tab.
3. Do one of the following:
  - To create a new alert, select Resources as the Alert Category, select the Alert Type, and then click Create Alert.

 To find out more about each alert type, select it to display a description on the right.

- To edit an existing alert, click the alert name.
4. In the Alert Information section:
    - a. Enter a unique name.
    - b. If you want to disable the alert, clear the Active check box.
    - c. Select the execution interval.

The execution interval specifies how often the alert runs and the amount of data that DPA examines. For example, if the execution interval is 10 minutes, DPA executes the alert every 10 minutes and examines the last 10 minutes of data to determine whether to trigger the alert. DPA recommends an execution interval of **at least 10 minutes** to prevent unnecessary alerts from a temporary condition.

- d. Enter the notification text to be sent with the email notification. Include an explanation of the issue and the suggested resolution.

 If you apply a custom [email template](#) to this alert, the email notification includes the notification text if the email template contains one of the following variables:

- Alert Notification text: [=alert.notificationText]
- Results: [=dpa.body] (included by default)

5. Select the database instances that the alert applies to.

6. Under Alert Parameters:

- a. For alerts against all metrics in a category, select the Category.

For alerts against a single resource metric, you can (optionally) select a Category to filter the Resource list.

- b. For alerts against a single resource metric, select the Resource.

- c. Specify the Calculation that is used to determine the alert level for an execution interval.

To determine the alert level, DPA looks at the values collected during an execution interval and applies the specified calculation.

Example: For a single resource alert, if the metric value is collected once each minute and the execution interval is 10 minutes, DPA looks at the 10 values collected for an interval

and applies one of the following calculations.

Calculation	Description
% meeting metric alarm criteria	<p>The alert is triggered when a certain percentage of the values collected during an execution interval meet or exceed the warning or critical threshold for the metric or metrics. Use the Percentage drop-down to specify the percentage.</p> <p>Example: The alert is for a category that contains 10 metrics. DPA collects the metric values once each minute, and the alert is set to run once every 10 minutes. Therefore, during each interval, DPA collects 100 values. If the Percentage is 75%:</p> <ul style="list-style-type: none"> <li>• The alert is triggered at the warning level if 75 or more of these values exceed the warning threshold for the associated metric, but fewer than 75 exceed the critical threshold.</li> <li>• The alert is triggered at the critical level if 75 or more of these values exceed the critical threshold for the associated metric.</li> </ul>
Average	DPA uses the average of the values collected during an interval to assign the alert level for that interval.
Median	DPA uses the median value collected during an interval to assign the alert level for that interval.
Maximum	DPA uses the maximum value collected during an interval to assign the alert level for that interval.
Minimum	DPA uses the minimum value collected during an interval to assign the alert level for that interval.

Category  Select a category to filter resource list

Resource  Only resources with defined alarm thresholds are available

Calculation  Applied against the metric data collected over the execution

Percentage  Percentage to use when selected calculation is '% meeting alarm criteria'

7. Select the person or group who gets notified when each alert level is triggered and when the

alert is broken. (The alert status is set to Broken if an error occurs during execution.)

To send notifications when the alert returns to Normal, select a recipient for Normal.

- If you have not added the person or group as a contact in DPA, click Add Contact or Add Contact Group and [create the contact or group](#).
- Select an [SNMP contact](#) to send SNMP traps when the alert is triggered. Select an SNMP contact for Normal to send a clearing notification when the alert status returns to Normal.

8. Verify or change the [notification policy](#).

- To send notifications when the alert returns to Normal, the notification policy must be Notify when level changes.

9. Select the [email template](#) that defines the contents of the email notifications sent by this alert.

10. Click Email Preview to see an example of the email that will be generated using the selected email template and contact information.

If the alert applies to multiple database instances, select an instance in the Email Preview dialog box and click OK. After reviewing the email, you can select a different database instance or click Cancel to close the Email Preview dialog box.

- The email sent to users might not exactly match the preview because some alert parameters cannot be evaluated during a preview.

11. Click Test Alert to test the alert and view the current alert level. The test does not generate an email.

12. Click Save.

## Configure a DPA Administrative alert


Administrative alerts are used to monitor the health of the database system. For example, you can configure an alert that is triggered when a database instance is not accessible or when any database parameter changes.

- For information about other DPA alert categories, see [DPA alerts](#).

1. From the DPA main menu, click Alerts.
2. Click the Manage Alerts tab.

### 3. Do one of the following:

- To create a new alert, select Administrative as the Alert Category, select the Alert Type, and then click Create Alert.

 To find out more about each alert type, select it to display a description on the right.


- To edit an existing alert, click the alert name.

### 4. In the Alert Information section:

- a. Enter a unique name.
- b. If you want to disable the alert, clear the Active check box.
- c. Select the execution interval.

The execution interval specifies how often the alert runs and the amount of data that DPA examines. For example, if the execution interval is 10 minutes, DPA executes the alert every 10 minutes and examines the last 10 minutes of data to determine whether to trigger the alert. DPA recommends an execution interval of **at least 10 minutes** to prevent unnecessary alerts from a temporary condition.

- d. Enter the notification text to be sent with the email notification. Include an explanation of the issue and the suggested resolution.


 If you apply a custom [email template](#) to this alert, the email notification includes the notification text if the email template contains one of the following variables:

- Alert Notification text: [=alert.notificationText]
- Results: [=dpa.body] (included by default)

### 5. Select the database instances that the alert applies to.





### 6. If any Alert Parameters are required for the alert type, enter the required value.

### 7. Specify the thresholds for each alert level you want to enable.

 Some Administrative alerts have only one level.


- Leave the Max value for the highest level **blank** to alert on anything above the minimum value for that level.
- If you configure multiple levels, the Max value for lower levels must **equal** the Min value for the next higher level.
- When you enter a Max value for a level, DPA alerts at that level when the value is **greater**

**than or equal to** the Min value but **less than** the Max level. For example, if the Min value is 5 and the Max value is 10, DPA will alert at that level when the value is 5 or when the value is 9.99, but **not** when the value is 10.


	Min (occurrences)	Max (occurrences)	
 HIGH	<input type="text" value="10"/>	<input type="text"/>	<div style="border: 1px solid black; padding: 5px;">A high-level alert is triggered when the value is 10 or greater.</div> <div style="border: 1px solid black; padding: 5px;">A medium-level alert is triggered when the value is 5 to anything <b>less than</b> 10.</div>
 MEDIUM	<input type="text" value="5"/>	<input type="text" value="10"/>	
 LOW	<input type="text"/>	<input type="text"/>	
 INFO	<input type="text"/>	<input type="text"/>	

- Select the person or group who gets notified when each alert level is triggered and when the alert is broken. (The alert status is set to Broken if an error occurs during execution.)

To send notifications when the alert returns to Normal, select a recipient for Normal.


-  If you have not added the person or group as a contact in DPA, click Add Contact or Add Contact Group and [create the contact or group](#).
- Select an [SNMP contact](#) to send SNMP traps when the alert is triggered. Select an SNMP contact for Normal to send a clearing notification when the alert status returns to Normal.

- Verify or change the [notification policy](#).

-  To send notifications when the alert returns to Normal, the notification policy must be Notify when level changes.

- Select the [email template](#) that defines the contents of the email notifications sent by this alert.
- Click Email Preview to see an example of the email that will be generated using the selected email template and contact information.

If the alert applies to multiple database instances, select an instance in the Email Preview dialog box and click OK. After reviewing the email, you can select a different database instance or click Cancel to close the Email Preview dialog box.


-  The email sent to users might not exactly match the preview because some alert parameters cannot be evaluated during a preview.

- Click Test Alert to test the alert and view the current alert level. The test does not generate an email.
- Click Save.



## Configure a DPA Custom alert

Use Custom alerts to execute SQL statements or stored procedures against the monitored database or DPA repository to check for conditions not covered by other DPA alerts. Each SQL statement or procedure returns a number (or set of numbers) that can trigger an alert depending on user-defined thresholds. Custom alerts can be used to alert against a wide variety of conditions. Any parameter that can be returned to DPA using a SQL statement or stored procedure can be used as the basis for a custom alert.

 For information about other DPA alert categories, see [DPA alerts](#).

To create a custom alert, see the following sections:

- [Custom alert types and expected return values](#)
- [Requirements for stored procedures](#)
- [Create or edit a Custom alert](#)
- [Custom tags](#)
- [Example: Create a custom DPA alert to display the name of the active node in a SQL Server cluster](#)

 Other examples of custom alerts can be found on [THWACK](#).

### Custom alert types and expected return values

Depending on what type of custom alert you select, the SQL statement or stored procedure must return one of the following values.

Alert type	Expected return values						
Single Numeric Return	The SQL statement or stored procedure returns a single numeric value. The alert is triggered if the value exceeds the defined High, Medium, Low, and Info thresholds.						
Multiple Numeric Return	<p>(SQL statements only.) The SQL statement returns one or more rows of data. Each row contains a string in the first column and a numeric value in the second column. For example, the query could return database names and the amount of free space for each one:</p> <table><tbody><tr><td>DB1</td><td>120</td></tr><tr><td>DB2</td><td>840</td></tr><tr><td>DB2</td><td>35</td></tr></tbody></table> <p>The alert is triggered if any value exceeds the defined High, Medium, Low, and Info thresholds.</p>	DB1	120	DB2	840	DB2	35
DB1	120						
DB2	840						
DB2	35						

Alert type	Expected return values
Single Boolean Return	The SQL statement or stored procedure returns a string value of <code>TRUE</code> or <code>FALSE</code> (not case-sensitive). The alert is triggered if <code>TRUE</code> is returned.
Single Alert Status Return	The SQL statement or stored procedure returns a string value that specifies the alert status. Valid values are <code>NORMAL</code> , <code>INFO</code> , <code>LOW</code> , <code>MEDIUM</code> , and <code>HIGH</code> (not case-sensitive).

## Requirements for stored procedures

When you create a custom alert that calls a stored procedure, the stored procedure must include **two** output parameters. These output parameters must be in the following order relative to each other, and no other output parameters can be included:

1. `AlertValue OUT VARCHAR2`

The value of this parameter must be one of the expected return values for the selected alert type. (For example, if the alert type is Custom Procedure Alert - Single Boolean Return, this output parameter must be `TRUE` or `FALSE`.)

Use the custom tag `#ALERTVALUE#` to include this output parameter.

2. `AlertString OUT VARCHAR2`

The value of this parameter is a description of the result of the stored procedure.

Use the custom tag `#ALERTSTRING#` to include this output parameter.

The stored procedure can include any number of input parameters. The input parameters can be interspersed with the output parameters, as long as the output parameters are in the correct order relative to each other. For example:


```
myproc('inputParam1', #ALERTVALUE#, 'inputParam2', #ALERTSTRING#, '#DBLINK#')
```

## Create or edit a Custom alert

1. From the DPA main menu, click Alerts.
2. Click the Manager Alerts tab.

3. Do one of the following:


- To create a new alert, select Custom as the alert category, select the alert type, and then click Create Alert.

 To find out more about each alert type, select it to display a description on the right.

- To edit an existing alert, click the alert name.

4. In the Alert Information section:

- a. Enter a unique name.
- b. If you want to disable the alert, clear the Active check box.
- c. Select the execution interval. (DPA recommends an execution interval of at least 10 minutes.)
- d. Enter the notification text to be sent with the email notification. Include an explanation of the issue and the suggested resolution.

 If you apply a custom [email template](#) to this alert, the email notification includes the notification text if the email template contains one of the following variables:

- Alert Notification text: [=alert.notificationText]
- Results: [=dpa.body] (included by default)

5. To run the SQL statement or stored procedure against monitored database instances (instead of the DPA repository), select the database instances.

6. In the Alert Parameters section:

- a. Enter the SQL statements to execute, or enter a call to a stored procedure.

Use [custom tags](#) to include variables such as the database ID and to include the required output parameters for stored procedures.

- b. In the Execute Against drop-down, indicate if the SQL statement or stored procedure should be executed against the selected database instances or against the DPA repository database.
- c. If the Description field is available, you can enter a custom description for the alert. This description replaces the DPA default description for the alert type when the Description parameter is included in the email template.
- d. If the alert returns a numeric value, specify the Units for the returned value.

7. If the alert returns a numeric value, specify the thresholds for each alert level you want to enable.
  - Leave the Max value for the highest level **blank** to alert on anything above the minimum value for that level.
  - If you configure multiple levels, the Max value for lower levels must **equal** the Min value for the next higher level.
  - When you enter a Max value for a level, DPA alerts at that level when the value is **greater than or equal to** the Min value but **less than** the Max level. For example, if the Min value is 5 and the Max value is 10, DPA will alert at that level when the value is 5 or when the value is 9.99, but **not** when the value is 10.

	Min (occurrences)	Max (occurrences)	
❌ HIGH	<input type="text" value="10"/>	<input type="text"/>	<div style="border: 1px solid black; padding: 5px;">           A high-level alert is triggered when the value is 10 or greater.         </div>
⊗ MEDIUM	<input type="text" value="5"/>	<input type="text" value="10"/>	
⚠️ LOW	<input type="text"/>	<input type="text"/>	<div style="border: 1px solid black; padding: 5px;">           A medium-level alert is triggered when the value is 5 to anything <b>less than</b> 10.         </div>
ℹ️ INFO	<input type="text"/>	<input type="text"/>	

8. Select the person or group who gets notified when each alert level is triggered and when the alert is broken. (The alert status is set to Broken if an error occurs during execution.)

To send notifications when the alert returns to Normal, select a recipient for Normal.

- ℹ️


  - If you have not added the person or group as a contact in DPA, click Add Contact or Add Contact Group and [create the contact or group](#).
  - Select an [SNMP contact](#) to send SNMP traps when the alert is triggered. Select an SNMP contact for Normal to send a clearing notification when the alert status returns to Normal.

9. Verify or change the [notification policy](#).

- ℹ️ To send notifications when the alert returns to Normal, the notification policy must be `Notify when level changes`.

10. Select the [email template](#) that defines the contents of the email notifications sent by this alert.
11. Click Email Preview to see an example of the email that will be generated using the selected email template and contact information.

If the alert applies to multiple database instances, select an instance in the Email Preview dialog box and click OK. After reviewing the email, you can select a different database instance or click Cancel to close the Email Preview dialog box.

 The email sent to users might not exactly match the preview because some alert parameters cannot be evaluated during a preview.

12. Click Test Alert to test the alert and view the current alert level. The test does not generate an email.
13. Click Save.

## Custom tags

You can include the following custom tags in your SQL statements or stored procedure calls. DPA replaces these tags at runtime with the appropriate values.

Tag	Description
#DBID#	<p>The internal DPA ID for the monitored database instance.</p> <ul style="list-style-type: none"><li>• <b>Data type:</b> VARCHAR2 (50)</li><li>• <b>SQL statement usage example:</b> <pre>select mycol from mytable where dbid=#DBID#</pre></li><li>• <b>Stored procedure usage example:</b> <pre>myproc(..., #DBID#, ...)</pre></li></ul>
#DBLINK#	<p>A database link used to connect to an Oracle monitored database.</p> <ul style="list-style-type: none"><li>• <b>Data type:</b> VARCHAR2 (50)</li><li>• <b>SQL statement usage example:</b> <pre>select mycol from myschema.mytable@#DBLINK#</pre></li><li>• <b>Stored procedure usage example:</b> <pre>myproc(..., '#DBLINK#', ...)</pre></li></ul>

Tag	Description
#ALERTVALUE#	<p>(Stored procedures only.) The first required output parameter for stored procedures. It returns one of the expected values based on the alert type. It must appear in the parameter list before #ALERTSTRING#.</p> <ul style="list-style-type: none"> <li>• Data type: VARCHAR2 (500)</li> <li>• Stored procedure usage example:</li> </ul> <pre>myproc (... , #ALERTVALUE# , ... , #ALERTSTRING#)</pre>
#ALERTSTRING#	<p>(Stored procedures only.) The second required output parameter for stored procedures. It returns a description of the alert condition.</p> <ul style="list-style-type: none"> <li>• Data type: VARCHAR2 (4000)</li> <li>• Stored procedure usage example:</li> </ul> <pre>myproc (... , #ALERTVALUE# , ... , #ALERTSTRING#)</pre>
#FREQUENCY#	<p>The execution interval for the alert, in minutes.</p> <ul style="list-style-type: none"> <li>• Data type: NUMBER</li> <li>• SQL statement usage example:</li> </ul> <pre>select mycol from myschema.mytable@#DBLINK# where mydate &gt; SYSDATE - (#FREQUENCY#/1440)</pre> <ul style="list-style-type: none"> <li>• Stored procedure usage example:</li> </ul> <pre>myproc (... , #FREQUENCY# , ...)</pre>

## Example: Create a custom DPA alert to display the name of the active node in a SQL Server cluster

This example shows how to configure two alerts that work together to cause DPA to display the physical machine name of the active node in a SQL Server failover cluster. By default, DPA shows only the cluster name.

- The first alert gets the name of the active node.
- The second alert appends this name to the name of the cluster that DPA displays. If the active node changes, it updates the name and also notifies recipients that a failover has occurred.

## Task 1: Create an alert to get the physical machine name

This alert runs against a monitored database instance and retrieves the name of the physical machine that the instance is currently running on.

1. On the DPA menu, click Alerts.
2. Click the Manage Alerts tab.
3. Choose Custom as the category and Custom SQL Alert - Multiple Numeric Return as the type.

**Alert Category**

Wait Time  Resources  Administrative  Custom

**Alert Type:** Custom SQL Alert - Multiple Numeric Return ▼ **Create Alert**

4. Click Create Alert.
5. Enter a unique name and select the execution interval.

Set the execution interval based on the average frequency of failovers in your clustered environment. The execution frequency affects the accuracy of the machine names that DPA displays. In this example, the alert interval is 10 minutes. But if failovers occur infrequently, you might want to choose a longer interval.

**Type** Custom SQL Alert - Multiple Numeric Return


**Description** Executes a user-defined SQL statement that will return one or more name/numeric value pairs

**Alert Name**

**Active**

**Execution Interval**   ▼

6. In the Notification Text box, provide a description of the alert.

 This alert does not send notifications, but the description provides information for other users.

*Notification Text - Explanation or resolution steps to be sent with alert email*

This alert gets the physical machine name for the active node in a SQL server cluster. It is used with the Change Physical Machine Name to display and update the active node name on the [DPA](#) homepage. The execution interval for this alert must be smaller than the interval for the Change Physical Machine Name alert.

7. Under Database Instances, select the SQL Server cluster.

Available Database Instances	Selected Database Instances
DPA-SUSE-MYSQL56:3306 DPAMARIADB:3306 DPAORA11R2ST_DPAORA11R2-STAN DPAORA11R2_DPAORA11R2 DPASQL2K14-BI DPASQL2K14-CS DPASQL2K16-RC3 DPASQL2K8 DPASQL2K8R2-WRG DPASY155:5000 DPASY155:5000	DPASQL2K12
<input type="button" value="Add"/>	<input type="button" value="View All"/>
<input type="button" value="Remove"/>	

8. Enter the following SQL in the SQL Statement box:

```
select coalesce(SERVERPROPERTY('ComputerNamePhysicalNetBIOS'),
SERVERPROPERTY('MachineName')) HOST, 0
```

\* SQL Statement - Enter a SQL statement that returns one or more rows containing a string in the first column and a numeric in the second column. The values will be used to evaluate the appropriate alert level.

```
select coalesce(SERVERPROPERTY('ComputerNamePhysicalNetBIOS'), SERVERPROPERTY('MachineName')) HOST,
0
```

9. Verify that Monitored Database is selected from the Execute Against drop-down menu.

10. Enter Machine Name 0 in the Units box.

Execute Against	<input type="button" value="Monitored Database"/>	<i>Determines whether Alert SQL will run against the Repository or the Monitored Database</i>
Description - Description of alert to be displayed in the alert notifications		
<input type="text" value="Physical Machine Name for Cluster"/>		
Units	<input type="text" value="Machine Name 0"/>	

11. Under Configure Alert Levels and Recipients:

- a. Enter 1 as the Min value for the High level.

The query returns a numeric value of 0. Entering 1 as the threshold ensures that the status is always Normal. Because this alert only retrieves the machine name, it should not be triggered.

- b. Do not select a contact because no one needs to receive notifications.



Configure Alert Levels and Recipients			
	Min	Max	Notification Group or Contact
<input checked="" type="checkbox"/> HIGH	<input type="text"/>	<input type="text"/>	-- Select a recipient --
<input checked="" type="checkbox"/> MEDIUM	<input type="text"/>	<input type="text"/>	-- Select a recipient --
<input checked="" type="checkbox"/> LOW	<input type="text"/>	<input type="text"/>	-- Select a recipient --
<input checked="" type="checkbox"/> INFO	<input type="text"/>	<input type="text"/>	-- Select a recipient --
<input checked="" type="checkbox"/> NORMAL			-- Select a recipient --
<input checked="" type="checkbox"/> BROKEN			-- Select a recipient --

12. Click Save.

## Task 2: Create an alert to append the machine name to the cluster name

This alert runs against the DPA Repository database. It appends ' Node: @nodeName' to the database instance name that DPA displays. Each time it runs, it determines whether the node name has changed. If so, it updates the display name and sends a notification so that you can investigate why the failover occurred.

This alert can also determine why the name has changed with different error levels. The error levels are: 0=no change, 1=node change, 2=initial update.

1. From the Manage Alerts tab, choose Custom as the category and Custom SQL Alert - Multiple Numeric Return as the type.

<b>Alert Category</b>	
<input type="radio"/> Wait Time	<input type="radio"/> Resources
<input type="radio"/> Administrative	<input checked="" type="radio"/> Custom
<b>Alert Type:</b>	Custom SQL Alert - Multiple Numeric Return
	<b>Create Alert</b>

2. Click Create Alert.
3. Enter a unique name and select the execution interval.

This execution interval must be **larger** than the interval for the Get Physical Machine Name alert. The execution intervals for both alerts affect the accuracy of the name that DPA displays. In this example, the execution intervals are 10 minutes and 12 minutes, but you should determine what intervals are appropriate for your environment.

Type	Custom SQL Alert - Multiple Numeric Return
Description	Executes a user-defined SQL statement that will return one or more name/numeric value pairs
Alert Name	<input type="text" value="Change Physical Machine Name for a Cluster"/>
Active	<input checked="" type="checkbox"/>
Execution Interval	<input type="text" value="12"/> <input type="text" value="Minutes"/>

- Enter the email notification text.

Notification Text - Explanation or resolution steps to be sent with alert email

The active node for the DPASQL2K12 cluster has changed. Investigate the reason for the failover.

- Under Database Instances, select the SQL Server cluster.

**i** You must select the SQL Server cluster for both alerts. If you do not select it for this alert, the instance name is not updated in DPA. If you do not select it for the Get Physical Machine Name alert, that alert does not run and has a status of Broken.

Available Database Instances	Selected Database Instances <span>View All</span>
<ul style="list-style-type: none"> <li>DPASUSE-MYSQL56:3306</li> <li>DPAMARIADB:3306</li> <li>DPAORA11R2ST_DPAORA11R2-STAN</li> <li>DPAORA11R2_DPAORA11R2</li> <li>DPASQL2K14-BI</li> <li>DPASQL2K14-CS</li> <li>DPASQL2K16-RC3</li> <li>DPASQL2K8</li> <li>DPASQL2K8R2-WRG</li> <li>DPASY155:5000</li> <li>DPASQL2K14-RC3</li> </ul>	<ul style="list-style-type: none"> <li>DPASQL2K12</li> </ul>
<input type="button" value="Add"/>	
<input type="button" value="Remove"/>	

- Enter the following SQL in the SQL Statement box:

```
declare
@mach_name varchar(100),
@current_name varchar(100),
@update_flag smallint

begin
select @mach_name=c.name
```

```
from con_alert_db a, con_alert b, con_alert_db_results c
where b.id = a.alertid
and b.alertname = 'Get Physical Machine Name'
and a.alertid = c.ALERTID
and a.DBID = c.DBID
and a.DBID = #DBID#

if @mach_name is not null
    select @update_flag = CHARINDEX(' Node',NAME)
    from COND
    where ID = #DBID#

if @update_flag != 0
    begin
        select @current_name = substring(NAME, CHARINDEX(' Node',NAME)
+7, 100)
        from cond
        where ID = #DBID#

        if ltrim(rtrim(@current_name)) != ltrim(rtrim(@mach_name))
            begin
                update COND
                set NAME = substring(NAME, 1,CHARINDEX(' Node',NAME)-1 )
                where ID = #DBID#

                update COND
                set NAME = NAME + ' Node: ' + @mach_name
                where ID = #DBID#

                select @mach_name,1
            end
        end
    else
        begin
            update COND
            set NAME = NAME + ' Node: ' + @mach_name
            where ID = #DBID#

            select @mach_name,2
```

```


end;
select @mach_name, 0
end;







```

7. Select Repository from the Execute Against drop-down menu.
8. Enter :Node|Error Level in the Units box.

Execute Against	<input type="text" value="Repository"/>	<i>Determines whether Alert SQL will run against the Repository or the Monitored Database</i>
Description - Description of alert to be displayed in the alert notifications		
Adding physical machine name to the instance cluster name. Error levels: 0=no change, 1=node change, 2=initial update		
Units	<input type="text" value=":Node Error Level"/>	

9. Under Configure Alert Levels and Recipients:
  - a. Enter 1 as the Min value for the High level.  
When the name of the active node changes, this alert is triggered at the High level.
  - b. Select a contact to receive the email when this alert is triggered.

 If you have not added the person or group as a contact in DPA, click Add Contact or Add Contact Group. See [Create contacts and contact groups](#).

Configure Alert Levels and Recipients			
	Min	Max	Notification Group or Contact
 HIGH	<input type="text" value="1"/>	<input type="text"/>	<input type="text" value="(GROUP)On Call"/>
 MEDIUM	<input type="text"/>	<input type="text"/>	-- Select a recipient --
 LOW	<input type="text"/>	<input type="text"/>	-- Select a recipient --
 INFO	<input type="text"/>	<input type="text"/>	-- Select a recipient --
 NORMAL	<input type="text"/>	<input type="text"/>	-- Select a recipient --
 BROKEN	<input type="text"/>	<input type="text"/>	-- Select a recipient --

10. Click Save.

## Send SNMP traps from DPA alerts

You can configure DPA alerts to send SNMPv2c traps to an SNMP-enabled Network Management System (NMS) when an alert level is reached and when the alert level returns to Normal. The trap contains the name of the monitored database instance, alert name, alert level, and response instructions.

To configure an alert to send an SNMP trap, complete the following tasks:

1. Import the [DPA MIB file](#) into your NMS.
2. [Create one or more SNMP contacts](#). The SNMP contact defines the response instructions included in the trap, and so you must create different contacts for alerts with different response instructions.
3. In the alert definition, add the SNMP contact as recipient for the alert level that you want to send a trap.

### The DPA MIB file

DPA contains a Management Information Base (MIB) file that defines the trap and the associated data sent with each trap. The MIB file defines the following:

- Private Enterprise Number
- One Trap Definition (NOTIFICATION-TYPE)
- Four string objects bound to each trap: database name, alert name, alert level, and response instructions

Before configuring DPA to send SNMP traps, provide the MIB file to the person responsible for importing MIB files into the NMS. The MIB file is in the following location:

```
<DPA_install_dir>/iwc/CONFIO-MIB.mib
```

### Create an SNMP contact

The NMS that receives the trap is represented as an SNMP contact in DPA.

1. On the DPA menu, click Options.
2. Under Administration > Users & Contacts, click Contact Management.
3. Click Create SNMP Contact.
4. Enter a name and description to identify the associated alert(s).

**i** By default, contacts are Active. You can select Inactive to disable the contact. When a contact is disabled, alerts associated with this contact do not send traps to the NMS.

5. Identify the NMS to send the trap to:
  - a. In the Trap Receiver Host field, enter the hostname or IP address of the server where the NMS is running.
  - b. In the Trap Receiver Port field, enter the port number where the NMS host is receiving traps. The default is 162.
  - c. In the Community String field, enter the community string used by the NMS for traps.
6. Enter the response instructions to be included in the trap.
7. To test the configuration, click Send Test SNMP Trap, and then verify that the NMS received the trap.
8. (Optional) To add the contact to a group, select the group and click Add.

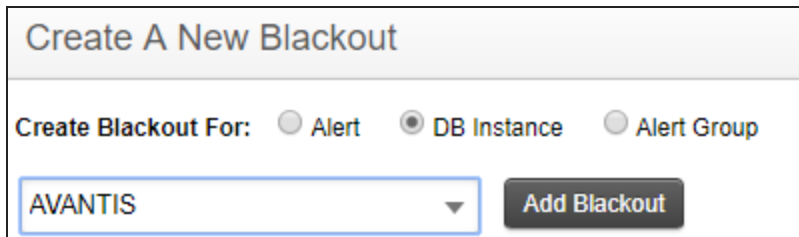
For example, when an alert reaches a certain level, you might want to send an email to the on-call personnel and send a trap to the NMS. You can add the SNMP contact to the On Call group.

## Stop DPA alerts for a period of time

To stop alerting for a period of time, create an alert blackout. For example, you can create an alert blackout to suppress alerts during a maintenance window.

### Create an alert blackout

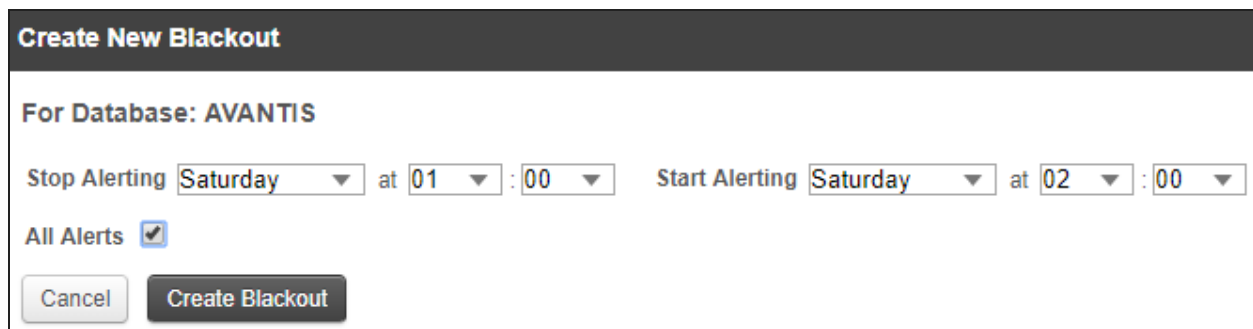
1. Click Alerts, and then click the Alert Blackouts tab.
2. Specify whether you want to create a blackout for an alert, a database instance, or an [alert group](#).
3. Select the alert, database instance, or alert group, and then click Add Blackout.



4. Specify the beginning and end of the blackout period. Times are based on a 24-hour clock.
5. If the blackout period is for an alert, select All Databases or specify the database instances that

this alert should not run on.

If the blackout period is for a database instance, select All Alerts or specify the alerts that should not run.



**Create New Blackout**

**For Database: AVANTIS**

Stop Alerting  at  :  Start Alerting  at  :


All Alerts

6. Click Create Blackout.

The blackout is effective each week on the specified day and time until it is deleted.

## Edit an alert blackout

1. Click Alerts, and then click the Alert Blackouts tab.
2. Under Existing Blackout Periods, locate the blackout period and click Edit.
3. Update the schedule and the affected alerts or database instances.

 You cannot change the original target of the blackout period. For example, if the blackout period was created to suppress all alerts on a database instance, you cannot change it to suppress a specific alert on multiple database instances. To make that type of change, delete the blackout period and create a new one.

4. Click Save Blackout.

## Delete an alert blackout

1. Click Alerts, and then click the Alert Blackouts tab.
2. Under Existing Blackout Periods, locate the blackout period and click Delete.
3. At the confirmation prompt, click Yes.

## Create a DPA alert group

An alert group defines a set of alerts to be run against a set of database instances. Alert groups simplify alert configuration and help make alerting more consistent across the monitored database instances. When you add alerts to an alert group, you do not have to select database instances within each alert definition. Instead, you select the database instances just once for the entire group. If the list of instances changes, you can update it in only one place.

1. On the DPA menu, click Alerts.
2. Click the Alert Groups tab.
3. Click Create Alert Group.
4. Enter a unique name and a brief description.

**Alert Group Information**

<b>Group Name</b>	<input type="text" value="Total Wait Times"/>
<b>Description</b>	<input type="text" value="Total blocking wait time and total DB instance wait time -- all database instances."/>

5. Select the alerts to include in this group.

Available Alerts		Selected Alerts
<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> MySQL InnoDB Buffer Pool Utilization Alert <small>For MySQL</small></li> <li><input checked="" type="checkbox"/> MySQL InnoDB Log File Size Alert <small>For MySQL</small></li> <li><input checked="" type="checkbox"/> SQL Server Deadlocks <small>For SQL Server</small></li> <li><input checked="" type="checkbox"/> Transaction Log Freespace for DPASQL2016 <small>For SQL Server, Sybase, DB2</small></li> <li><input type="checkbox"/> Transaction Log Freespace on DPASQI 2K12</li> </ul>	<input type="button" value="Add"/>  <input type="button" value="Remove"/>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Total Blocking Wait Time <small>For all DB Instance Types</small></li> <li><input checked="" type="checkbox"/> Total Database Instance Wait Time <small>For all DB Instance Types</small></li> </ul>

6. Select the database instances on which to execute these alerts.

Available Database Instances		Selected Database Instances
<div style="border: 1px solid gray; height: 100px; width: 100%;"></div>	<input type="button" value="Add"/>  <input type="button" value="Remove"/>	<ul style="list-style-type: none"> <li>DPA-SUSE-MYSQL56:3306 <small>MySQL</small></li> <li>DPAMARIADB:3306 <small>MySQL</small></li> <li>DPAORA11R2ST_DPAORA11R2-STAN <small>Oracle</small></li> <li>DPAORA11R2_DPAORA11R2 <small>Oracle</small></li> <li>DPASQI 2K12</li> </ul>

7. Click Save.



# Create and manage DPA contacts and contact groups

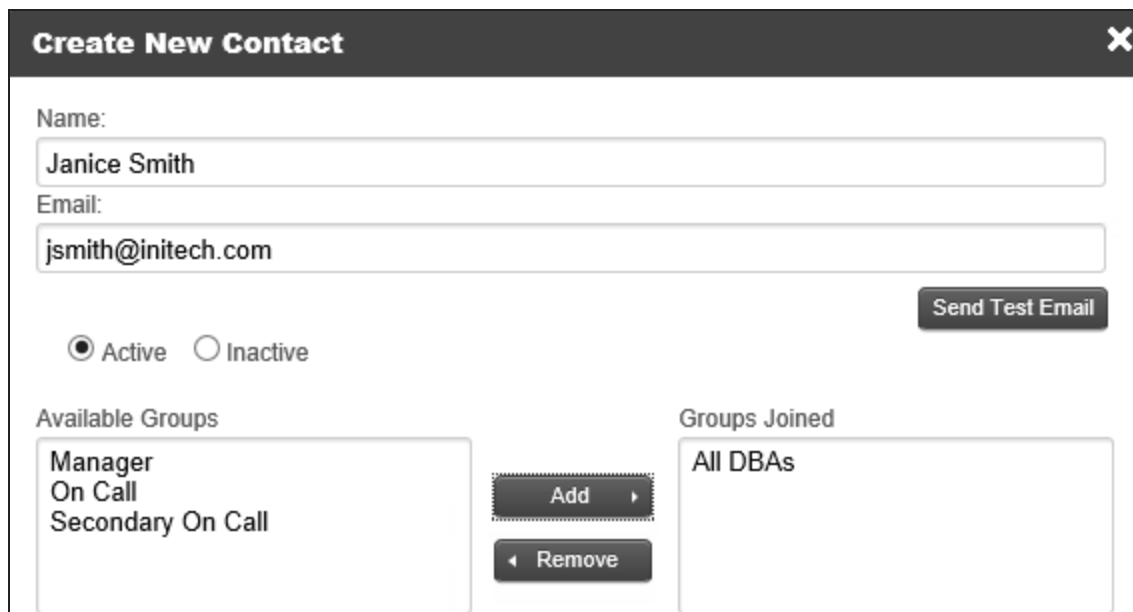
Before you create DPA [alerts](#) or schedule [reports](#), define the contacts and contact groups who can receive the alerts and reports. As your organization changes, you can edit or delete contacts or contact groups.

**i** If you want to send DPA alerts as SNMP traps to your Network Management System (NMS), [create an SNMP contact](#).

## Create a contact

Contacts are people who can receive email notifications when an alert is triggered, or who can receive scheduled reports through email. When you define an alert or schedule a report, you can select the recipient from the list of available contacts.

1. On the DPA menu, click Options.
2. Under Administration > Users & Contacts, click Contact Management.
3. In the Email Contacts section, click Create Contact.
4. Enter the contact's name and email address. Optionally, add the contact to an existing group.



**Create New Contact** ✕

Name:  
Janice Smith

Email:  
jsmith@initech.com

Active  Inactive

Send Test Email

Available Groups: Manager, On Call, Secondary On Call

Groups Joined: All DBAs

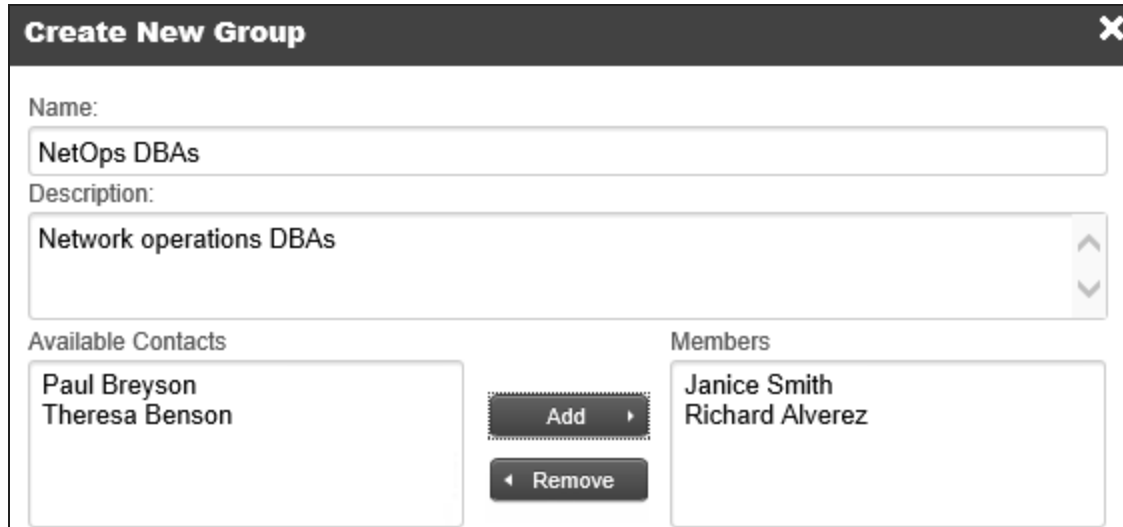
Add Remove

5. Click Save.

## Create a contact group

Contact groups are used to send emails to multiple people when an alert is triggered or when a scheduled report runs. DPA provides several default contact groups, but you can create other groups.

1. On the DPA menu, click Options.
2. Under Administration > Users & Contacts, click Contact Management.
3. In the Groups section, click Create Group.
4. Enter a group name and description. Optionally, add existing contacts to the group.



**Create New Group** [X]

Name:  
NetOps DBAs

Description:  
Network operations DBAs

Available Contacts: Paul Breyson, Theresa Benson

Members: Janice Smith, Richard Alvarez

Buttons: Add, Remove

5. Click Save.

## Edit a contact or contact group


1. On the DPA menu, click Options.
2. Under Administration > Users & Contacts, click Contact Management.
3. Click the name of the contact or contact group to open the Update Contact or Update Group dialog box.
4. Make the necessary changes and click Save.

## Delete a contact or contact group

1. On the DPA menu, click Options.
2. Under Administration > Users & Contacts, click Contact Management.
3. In the right column of the contact or group table, click Delete.
4. On the confirmation dialog, click Yes.

## Notification policy for DPA alerts


When you create an alert in DPA, you can accept the default notification policy or apply a different policy to that alert. The notification policy determines when notifications about the alert are sent. The following sections describe each policy.

 When an alert level changes, DPA sends a notification only if a recipient is selected for that level in the alert definition. For example, if the notification policy is `Notify when level changes`, you must specify a recipient for Normal if you want DPA to send a notification when the alert level returns to Normal. The examples in the following tables assume that recipients are specified for all alert levels.

### Notify when level not visited since normal

A notification is sent if the alert status is not Normal and the alert has not been in this status since the last time the status was Normal. If the alert returns the same status for multiple polling periods without returning to Normal, you are notified only once for each status. For example:

Execution Interval	Alert Level	Notification Sent?
1	Normal	No
2	Medium	Yes
3	High	Yes
4	High	No (this alert level was returned previously)
5	Medium	No (this alert level was returned previously)
6	Low	Yes
7	Normal	No
8	Low	Yes

 This is DPA's default notification policy. A DPA administrator can change the default policy for your DPA deployment by [setting the Advanced Option](#) `ALERT_NOTIFICATION_TRIGGER`.

## Notify when level changes

A notification is sent if the alert status has changed since the previous execution interval, even if the change is that it returned to Normal. You are notified each time the level changes, but only once for each change. For example:

Execution Interval	Alert Level	Notification Sent?
1	Normal	No (the alert was not triggered during the previous execution interval)
2	Medium	Yes
3	High	Yes
4	High	No (the alert status has not changed)
5	Medium	Yes
6	Low	Yes
7	Normal	Yes
8	Low	Yes

## Notify when level is not normal

A notification is sent if the alert status is not Normal, regardless of the alert's previous status. For example:

Execution Interval	Alert Level	Notification Sent?
1	Normal	No
2	Medium	Yes
3	High	Yes
4	High	Yes
5	Medium	Yes
6	Low	Yes
7	Normal	No
8	Low	Yes

# Define email templates for alert notifications

When an alert is triggered, DPA sends an email to notify the designated recipients. Email templates define the contents of the email notification. DPA provides a DPA System Template, but you can create custom templates and assign them to alert definitions:

- [Create or edit a custom email template for DPA alert notifications](#)
- [Delete a custom email template](#)
- [Change the default email template for DPA alert notifications](#)

 Admin privileges are required to create or manage custom email templates.


## Create or edit a custom email template for DPA alert notifications

Use custom [email templates](#) to customize the contents of the email notification that DPA sends when an alert is triggered. You can create multiple custom email templates for different types of alerts.

1. On the DPA menu, click Alerts.
2. Click the Email Templates tab.
3. Do one of the following:
  - To create a new email template, click Create email template.  
The Create email template page opens. It includes the system-defined template definition as a starting point.
  - To edit an existing email template, click the email template name.  
The Edit email template page displays the existing template definition.
4. Enter a unique name and, optionally, a description.


5. Specify the content and formatting of email notifications based on this template:

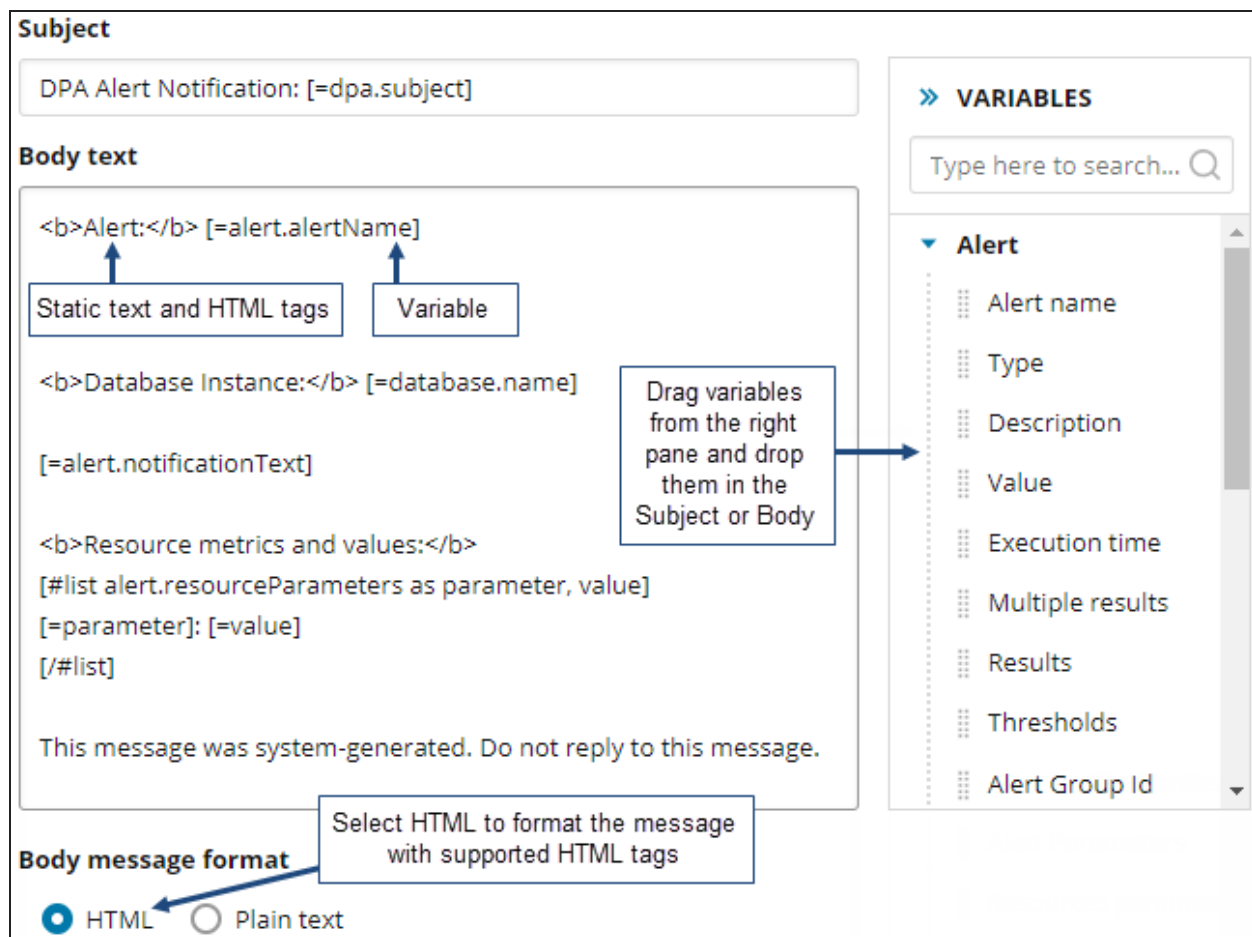
- To add content:
  - Drag and drop variables from the right panel into the Subject line or body (or type the variables). These variables represent information about the triggered alert or links to additional information in DPA. For information about the available variables, see [Email template variables](#).
  - Type static text into the Subject line or body.
- To remove content, delete variables or static text from the Subject line or body.
- To format body text with bold, italics, or line breaks:
  - a. Under Body message format, select HTML.

 If Plain text is selected, any HTML tags are treated as text and shown in the alert notification email.

b. Enter the following tags to format the body text:

```
<b> </b>  
<i> </i>  
<br> or <br />
```

 All other tags are unsupported. You cannot save a template that contains unsupported tags.



**Subject**

DPA Alert Notification: [=dpa.subject]

**Body text**

<b>Alert:</b> [=alert.alertName]

Static text and HTML tags    Variable

<b>Database Instance:</b> [=database.name]

[=alert.notificationText]

Drag variables from the right pane and drop them in the Subject or Body

<b>Resource metrics and values:</b>

[#list alert.resourceParameters as parameter, value]

[=parameter]: [=value]

[/#list]

This message was system-generated. Do not reply to this message.

**Body message format**

Select HTML to format the message with supported HTML tags

HTML     Plain text

**VARIABLES**

Type here to search... Q

▼ Alert

- Alert name
- Type
- Description
- Value
- Execution time
- Multiple results
- Results
- Thresholds
- Alert Group Id

- Click Save to save your changes and close the page.
- Apply the email template to alerts in either of the following ways:
  - To apply the template to a specific alert, [edit the alert definition](#) and select the template from the Email Template drop-down menu.
  - To apply the template to all alerts that use the default email template, [specify this template as the default](#).

## Email template variables

The following variables are available:

- [Alert variables](#)
- [Database variables](#)
- [Link variables](#)
- [DPA alert variables](#)
- Custom properties

## Alert variables

Use these variables to include information about the alert that was triggered.

Name	Variable	Description
Alert name	[=alert.alertName]	The user-defined name that identifies the alert that was triggered.
Type	[=alert.type]	The type of alert.
Description	[=alert.description]	DPA's description of the alert type.
Value	[=alert.value]	The value that triggered the alert.
Execution time	[=alert.executionTime]	The date and time when the alert was triggered.
Multiple results	[=alert.multiReturn]	A value of true or false to indicate whether the alert returns multiple values.
Results	[#list alert.results as result] [=result.category]: [#if result.parameterName??] [=result.parameterName] [#else] [/#if] [=result.label]: [=result.value] [#if result.units??] [=result.units][#else] [/#if] [=result.description] [/#list]	For Custom alerts that return multiple values, the parameter name, value, units (if specified in the alert parameters), and description (if specified in the alert parameters) for each returned value.
Thresholds	[#list alert.threshold as t]  Threshold level: [=t.level] * min: [#if t.levelMin??] [=t.levelMin][#else]N/A [/#if] * max: [#if t.levelMax??] [=t.levelMax][#else]N/A [/#if] [/#list]	The minimum and maximum values for threshold levels that are specified in the alert definition.

**i** This variable does not return threshold information for Resources alerts because the thresholds are defined on the resource instead of in the alert definition.



Name	Variable	Description
Alert Notification text	[=alert.notificationText]	The text from the Notification Text field in the alert definition.
Alert Parameters	[#list alert.alertParameters as parameter, value] [=parameter]: [=value] [/#list]	For Wait Time and Administrative alerts, the name and value of the each parameter specified for the alert. If the alert type does not require parameters, this is blank.
Resources parameters	[#list alert.resourceParameters as parameter, value] [=parameter]: [=value] [/#list]	For Resources alerts, the name and value of each parameter specified for the alert.
Alert Group ID	[=alert.group.id]	The ID of the alert group that the alert belongs to.
Alert Group Name	[=alert.group.name]	The name of the alert group that the alert belongs to.
Alert Group Description	[=alert.group.description]	The user-defined description of the alert group that the alert belongs to.
Alert Status Value	[=alert.status.value]	The status of the alert (for example, High or Broken).
Alert Error Message	[=alert.status.message]	When the alert is Broken, the error message generated when the alert is triggered.
Single Alert	[=alert.singleAlert]	A value of true or false to indicate whether multiple result values are sent in a single message. True indicates that multiple results are sent in one message, and false indicates that they are sent separately.

## Database variables

Use these variables to include information about the monitored database instance on which the alert was triggered.

Name	Variable	Description
Name	[=database.name]	The DPA display name of the monitored database instance.

Name	Variable	Description
Type	[=database.databaseType]	The type of monitored database instance.
IP address	[=database.ipAddress]	The IP address of the database server.
Hostname	[=database.hostname]	The host name of the database server.
Port	[=database.port]	The port used by the monitored database instance.
Version	[=database.databaseVersion]	Version of the monitored database instance.
Full type	[=database.databaseFullType]	The full type of the monitored database instance.

## Link variables

Use these variables to include links to related information in DPA.

Name	Variable	Description
Alert Status	[=links.alertStatus]	A link to the Alert Status tab, from which you can <a href="#">view the status and history of the alert</a> .
Alert Trends	[=links.alertTrends]	A link to the 1-day trends chart for the day on which the alert was triggered.
Alert History	[=links.alertHistory]	A link to the <a href="#">detailed history</a> of the alert on the database instance where it was triggered.
Instance Alerts	[=links.instanceAlerts]	A link to the Alert Status tab filtered to show only the alerts configured to run on the database instance where the alert was triggered.
Notification	[#list links.notification as label, url] [=label]: [=url] [/#list]	A link to the DPA chart associated with the alert type. For example, the link for a Database Instance Wait Time Anomaly alert opens the Anomaly Detection chart for the day when the alert was triggered. If no chart is associated with the alert type, this is blank.

## DPA alert variables

These variables define the default DPA alert content.

Name	Variable	Description
Subject	[=dpa.subject]	<p>The default subject line of an email alert. This variable includes the alert name, the database instance, and the alert level in the following format:</p> <pre>Alert Name (Database Instance) - ALERT LEVEL</pre> <p>For example:</p> <pre>Total SQL Wait Time for Memory/CPU Waits (MyDatabaseInstance) - HIGH</pre>
Results	[=dpa.body]	<p>The default body text of an email alert. This variable includes the alert status link, the alert notification text, and the value that triggered the alert. Each element is a separate paragraph. For example:</p> <pre>View Alert Status: http://xxxxxxxx:8123/iwc/alertMain.iwc</pre> <p>The total wait time for Memory/CPU waits has exceeded a threshold.</p> <pre>Value: 600 seconds</pre>

## Custom properties

You can use the [DPA REST API](#) to create custom properties that can be used in alert notification email templates. If any custom properties have been created for this DPA server, they are displayed in the list of variables.

## Delete a custom email template

If you delete a custom [email template](#) that is assigned to one or more alerts, DPA assigns the default template to those alerts.



- You cannot delete the DPA System Template.
- You cannot delete a custom template that is currently designated as the default template.

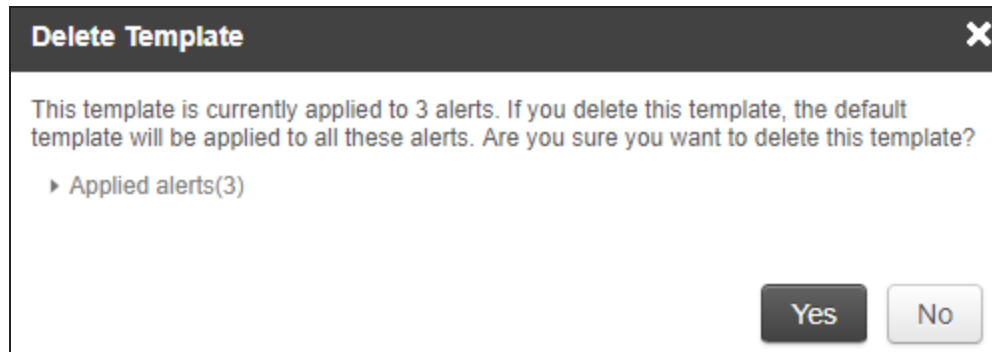
To delete a custom template:

1. On the DPA menu, click Alerts.
2. Click the Email Templates tab.

3. If the template you want to delete is currently designated as the default template, [designate a different template as the default template](#).
4. Locate the table row that shows the template you want to delete, and click Delete.

If the template is not assigned to any alerts, DPA displays a simple confirmation message.

If the template is assigned to one or more alerts, DPA displays the following confirmation message. To see which alerts use the template, click Applied alerts.

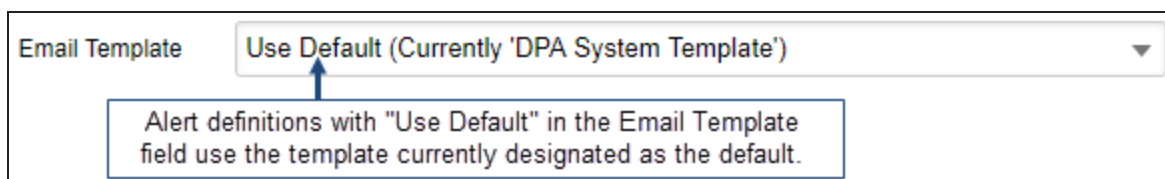


5. Click Yes at the confirmation message to delete the template.

Any alerts that previously used that template now use the default template.

## Change the default email template for DPA alert notifications

If an alert definition does not assign a specific [email template](#) to use for alert notifications, the default email template is used.



Initially, the DPA System Template is the default template. You can [create a custom template](#) and designate it as the default.

1. On the DPA menu, click Alerts.
2. Click the Email Templates tab.


"Default" next to the template name identifies the default template.

Email Templates			Create email template
Name ▲	Description	Applied Alerts	
ABC company template	This is the default template defined for the ABC Compan...	0	Make default Delete
Resource metric alerts template	This template defines the email sent when a resource me...	1	Make default Delete
DPA System Template <span>Default</span>	Default template, assigned to 6 alerts	6	

3. Locate the template you want to designate as the default, and click Make default. Then click Yes at the confirmation prompt.

"Default" appears next to the template's name, and the number of Applied Alerts is updated. All alert definitions with "Use Default" in the Email Template field now use this template for email notifications.

Email Templates			Create email template
Name ▲	Description	Applied Alerts	
ABC company template <span>Default</span>	This is the default template defined for the ABC Compan...	6	
Resource metric alerts template	This template defines the email sent when a resource me...	1	Make default Delete
DPA System Template		0	Make default

 The Delete button is no longer displayed for the new default template, because you cannot delete a template while it is designated as the default.

## DPA reports

To work with DPA reports, see the following topics:

- Learn [about the available report types](#) and the differences between report data and chart data.
- Access and [run existing DPA reports](#).
- Create a [new DPA report](#).
- [Search for a SQL statement](#) to include in a report.
- [Schedule](#) a DPA report for email delivery.
- Create a [DPA report group](#).

### About DPA reports

Use reports to communicate the long-term performance of your databases and supply evidence to support your work. Reports can capture the results of performance tuning and highlight database trends. You can send reports to managers, team members, and customers.

### Differences between report data and detailed chart data displayed in the DPA interface

The data shown in reports differs from the detailed chart data shown in the DPA interface.

	Report data	Detailed chart data
Storage period and granularity	<p>Reports can show data captured over longer intervals and display long-term trends. To generate reports, DPA summarizes repository data to make long-term information available in a manageable size.</p> <p>The previous 90 days of data are summarized by hour. After 90 days, data are summarized by day. This information is available for five years.</p>	<p>Detailed chart data are available for a shorter period, typically 30 days.</p> <p>Charts can show information down to the second.</p>
Data collection period	<p>Reports can be generated after a one-hour data collection period. SolarWinds recommends allowing a 24-hour data collection period before you create a report.</p>	<p>Charts display data after a 10-minute data collection period.</p>

## Report types

DPA has many standard reports that include the most commonly used wait time statistics. You can customize each report by selecting the database instance, time interval, and the items included (for example, the wait types or SQL statements). The following types of reports are available.

- Average Wait

Reports in this category show the average times for a single SQL statement or multiple SQL statements.

- Top <element>

These reports show the files, SQL statements, users, or other elements that are experiencing the longest waits. For example, the Top Files report shows the busiest files ranked by total I/O wait time.

- Typical Day of <element> Wait

These reports show the times of day when files, SQL statements, users, or other elements are experiencing the longest waits.

## Learn more

To work with DPA reports, see the following topics:

- [Access and run DPA reports](#)
- [Create a DPA report](#)
- [Schedule a DPA report for email delivery](#)
- [Create a DPA report group](#)

## Access and run DPA reports

From the Reports tab, you can view existing reports and [create new reports](#).

1. On the DPA menu, click Reports.

The Reports section lists the reports that have been created on this DPA server.

2. In the Reports section, you can:

- Choose a database instance from the drop-down menu in the upper-right corner to filter the list of reports.

- Click Show to run and open a report.
- Click a column heading to sort the list of reports.
- In the right column, click Delete to delete a report.

## Create a DPA report

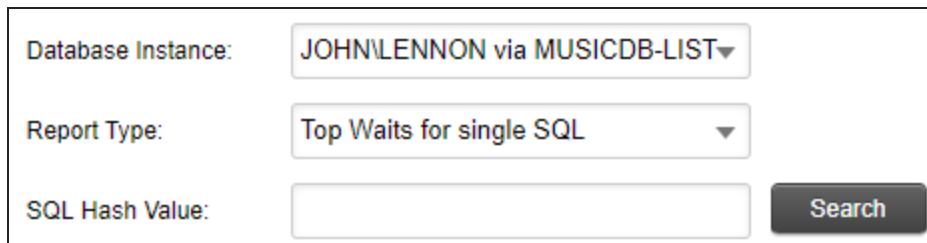
Use DPA reports to identify database trends and track the results of your performance tuning.

1. On the DPA menu, click Reports.
2. Select the Database Instance and the Report Type.



Database Instance: JOHN\LENNON via MUSICDB-LIST ▼  
Report Type: Top Waits for single SQL ▼

3. If the report type shows information about a **single** SQL, plan, or wait (for example, the Top Waits for single SQL report), identify the SQL, plan, or wait:
  - a. Click Search next to the field that is added to the Create a New Report pane.



Database Instance: JOHN\LENNON via MUSICDB-LIST ▼  
Report Type: Top Waits for single SQL ▼  
SQL Hash Value:  Search

- b. Locate the SQL, plan, or wait and click OK to add it.

To find a SQL statement, see [Search for a SQL statement to report on](#).

4. Click Report Options.

The Report - Advanced Options page opens.

5. Depending on the report type, specify which waits, SQL statements, or other elements to display in the report.

By default, the report includes the elements with the highest wait time. To include specific elements, select User-Defined, click Add, and then use the Search box to locate and add up to 50 elements.

To find a SQL statement, see [Search for a SQL statement to report on](#).



### Waits to Display


Top Waits Ranked by Cumulative Wait Time

Top  Waits

User-Defined Waits

db file async I/O submit  
 Disk file operations I/O  
 Log archive I/O

6. Under Dates to Display, specify the dates that the report should include.

 The Data Range at the bottom of this section shows the time period for which data is available.

### Dates to Display

Date Range:  Hour Range:  to

Days of Week:

Dates: **September 1, 2016 - September 30, 2016**

Data Range: July 26, 2016 - September 14, 2016

7. Under General, complete the following fields.

Report Name	Enter a unique name to identify this report in the report list.
Report Title	(Optional) Enter a title to display at the top of the report. If you leave this field blank, the report title defaults to the report type, database instance, and time period.
Report Description	(Optional) Enter a description to explain the report's content or purpose.

8. In the New Report section at the top of the window, click Display Report.

The report opens.

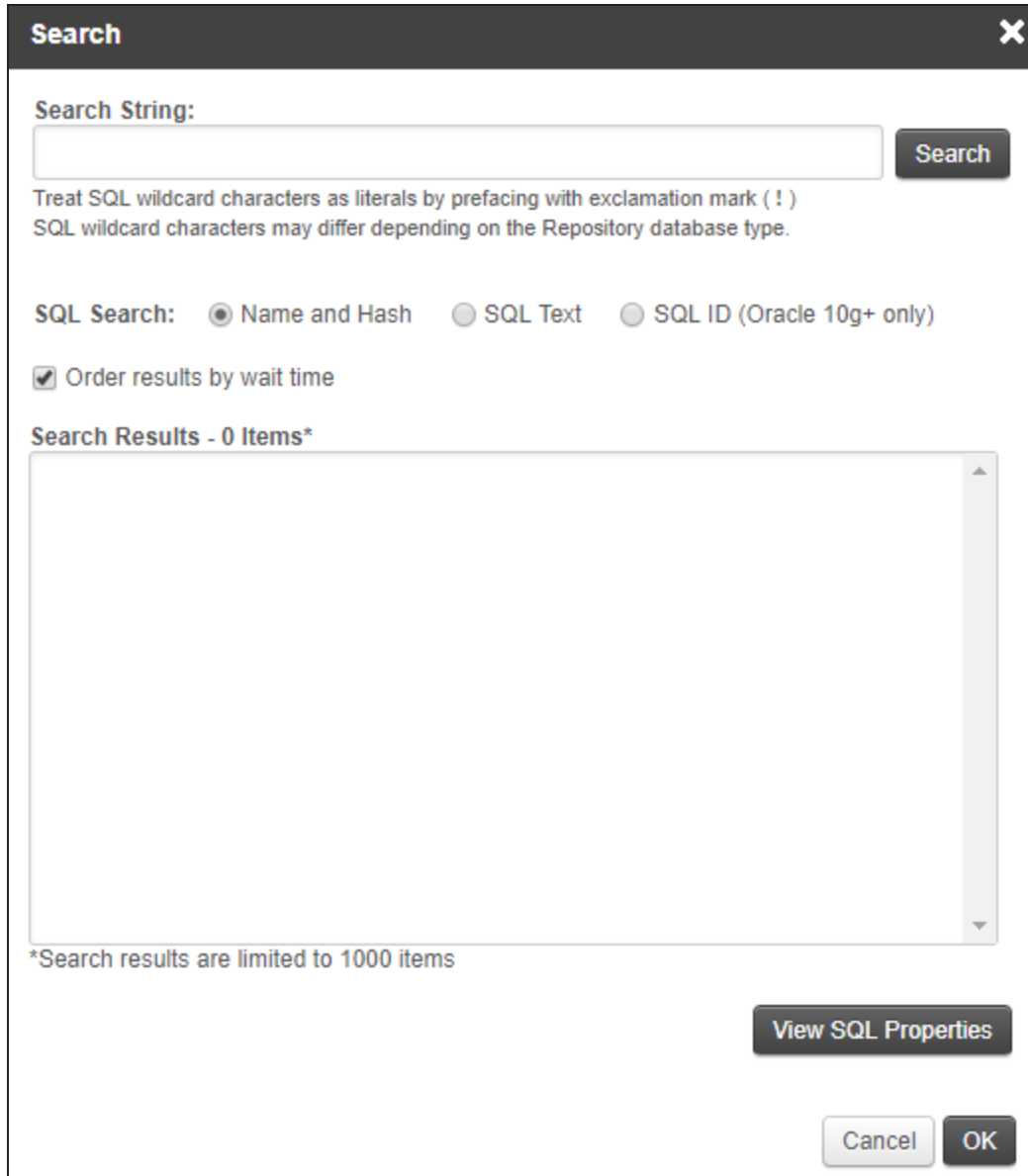
9. Choose one of the following options:

Click	If you want to
Save	Save the report with the name you entered previously.
Save As	Save the report with a different name.
Edit	Return to the Report - Advanced Options page and make changes.
Email Report	Send the report to one or more users.

You can view the report from the Reports tab at any time, or [schedule the report](#) to run automatically and be emailed to a group of recipients.

## Search for a SQL statement to report on

When you [create a report](#) to show information about a single SQL statement or a group of SQL statements, you must identify the SQL statements. When you click the Search button on the Reports page or the Select Items to Display dialog, the following Search dialog opens:



The image shows a 'Search' dialog box with a dark header and a close button (X) in the top right corner. Below the header, there is a 'Search String:' label followed by a text input field and a 'Search' button. Below the input field, there is a note: 'Treat SQL wildcard characters as literals by prefacing with exclamation mark (!) SQL wildcard characters may differ depending on the Repository database type.' Below this note, there are three radio buttons for 'SQL Search': 'Name and Hash' (selected), 'SQL Text', and 'SQL ID (Oracle 10g+ only)'. Below the radio buttons, there is a checked checkbox for 'Order results by wait time'. Below the checkbox, there is a large empty text area labeled 'Search Results - 0 Items\*'. At the bottom left of the text area, there is a note: '\*Search results are limited to 1000 items'. At the bottom right of the dialog, there are three buttons: 'View SQL Properties', 'Cancel', and 'OK'.

To search for SQL statements, complete the following steps.

1. Select a SQL Search option, and then enter a string in the Search String:

- **Name and Hash** (selected by default)

If you [named the SQL statement](#), enter part of the name. If not, enter part of the hash value that DPA uses to identify unnamed SQL statements.

- **SQL Text**

Enter a string that is included in the SQL statement. For example, entering `where` returns all SQL statements that include a `WHERE` clause (up to the limit of 1000 search results).

The search string can include SQL wildcard characters. For example, the following search string uses `%` as a substitute for 0 or more characters:

```
select%Project%where>Status
```

It returns all `SELECT` statements against a table named `Project` that include a column named `Status` in the `WHERE` clause.

- **SQL ID**

For SQL statements that run against an Oracle 10g or later database, enter part of the SQL ID.

2. Click Search.

By default, the Search Results box lists the name or hash value of each SQL statement, as well as the SQL statement's total wait time for the last seven days. The results are ordered by wait time with the highest waits first.

### Search ✕

Search String:

Treat SQL wildcard characters as literals by prefacing with exclamation mark (!)  
SQL wildcard characters may differ depending on the Repository database type.


SQL Search:  Name and Hash  SQL Text  SQL ID (Oracle 10g+ only)

Order results by wait time

Search Results - 840 Items\* (wait time format mm:ss)

4978808246	- 04:44
4710869972	- 04:29
3592459538	- 04:22
2809080563	- 04:00
2372187159	- 02:52
2901106198	- 02:39
5192629427	- 02:26
2557494638	- 02:17
2929200436	- 02:13
3980931161	- 02:07
4990026383	- 02:06
4660260622	- 02:04
5969189043	- 02:04
3334054254	- 02:00
3860811288	- 01:55
3581785588	- 01:18

\*Search results are limited to 1000 items

-  To change the amount of wait time shown in the search results, [set the advanced configuration option](#) REPORT\_SEARCH\_WAIT\_TIME\_DAYS.
- To list the search results in alphanumeric order without wait times, deselect Order results by wait time. Then click Search again.

3. Locate the SQL statement you want to report on, and select it.

For more information about any SQL statement, select it and click View SQL Properties to see the SQL text.

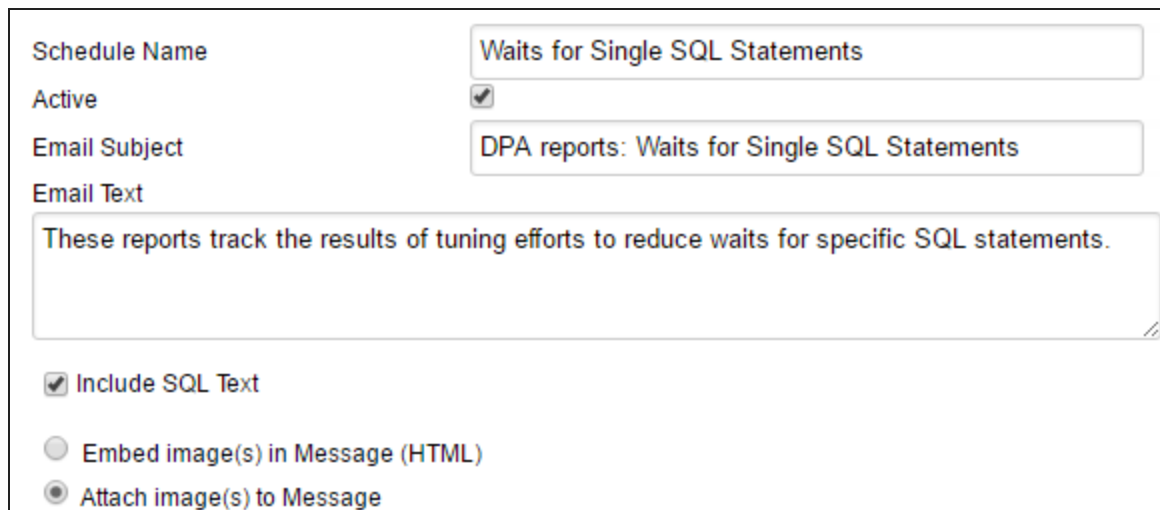
4. Click OK to add the SQL statement to the report.

## Schedule a DPA report for email delivery

Report schedules automatically email reports (or [report groups](#)) at regular intervals. You can send reports to managers, team members, and customers. Use report schedules to communicate database trends to people who do not have direct access to DPA, and to and highlight performance improvements across your organization.

- Only DPA administrators can create report schedules.
- The report recipients must be [added as DPA contacts](#).

1. Log in to DPA using an account with administrator privileges.
2. On the DPA menu, click Reports.
3. Click the Report Schedules tab.
4. Click Create Schedule.
5. Name the schedule and enter an email subject and body text.



Screenshot of the 'Create Schedule' form in SolarWinds DPA. The form includes the following fields and options:

- Schedule Name:** Waits for Single SQL Statements
- Active:**
- Email Subject:** DPA reports: Waits for Single SQL Statements
- Email Text:** These reports track the results of tuning efforts to reduce waits for specific SQL statements.
- Include SQL Text:**
- Embed image(s) in Message (HTML):**
- Attach image(s) to Message:**

6. Specify when you want the report delivered, and click Add. You can specify multiple delivery times.

Select the type of delivery pattern to create a delivery time. Multiple times are allowed.

Weekly Pattern

Day Of Week: Thursday

Delivery Time: 8 : 00 AM

Monthly Pattern

Day Of Month: 1

Delivery Time: 8 : 00 AM

Add Remove

Weekly - Monday at 8:00 AM  
Monthly - 1st at 8:00 AM

- Under Available Reports, select the reports or report groups, and click Add.

Available Reports

(GROUP) Auto-Email Reports  
(GROUP) Waits for Single SQL Statements  
DPA-SUSE-MYSQL56:3306 - Top SQL  
DPA-SUSE-MYSQL56:3306 - Top Waits  
DPAORA11PER\_DPAORA11PER - Top SQL  
DPAORA11PER\_DPAORA11PER - Top Waits  
DPASQL2K12 - Top SQL  
DPASQL2K12 - Top Waits  
DPASQL2K14-BI - Top SQL  
DPASQL2K14-BI - Top Waits

Add Remove

Selected Reports

(GROUP) Top Waits

- If you want to review the email that will be sent when the schedule runs, click Send Test Email and enter an email address.
- Under Available Contacts, select the recipients of the report, and click Add.

**i** If you have not [added the recipients as contacts](#) in DPA, click Add Contact or Add Contact Group.

Available Contacts

(GROUP) All DBAs  
(GROUP) Manager  
(GROUP) On Call  
(GROUP) Secondary On Call  
Janice Smith  
Paul Breyson  
Richard Alvarez  
Theresa Benson

Add Remove

Selected Contacts

(GROUP) NetOps DBAs

- Click Create Schedule.

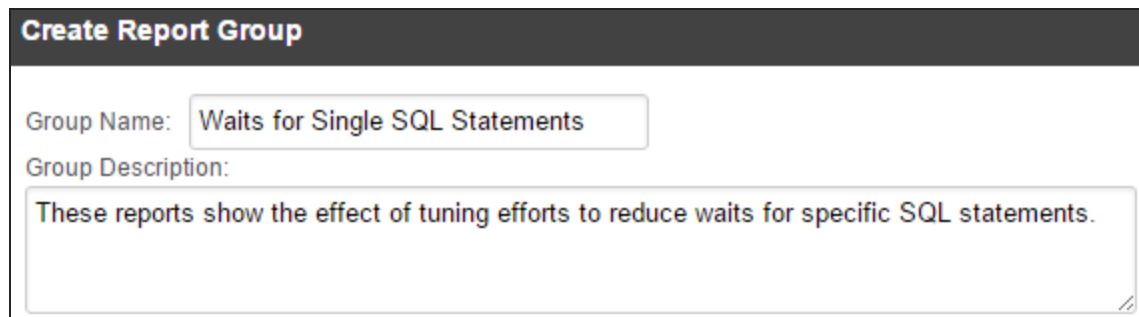
Your schedule is added to the list of report schedules.

Does your network or firewall require an internal SMTP server? If so, see [SMTP mail server for outgoing email](#).

## Create a DPA report group

Use report groups to display data from related reports on the same page. With report groups, you can quickly run or schedule multiple reports.

1. On the DPA menu, click Reports.
2. Click the Report Groups tab.
3. Click Create Report Group.
4. Give the group a name and (optionally) a description.

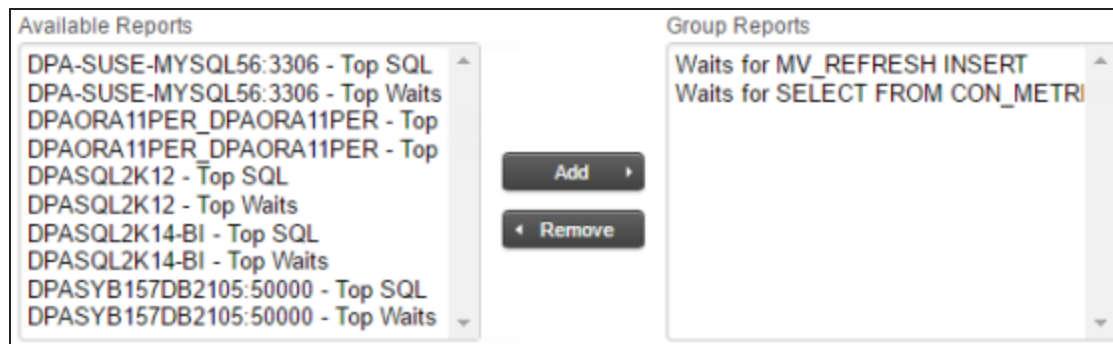


**Create Report Group**

Group Name:

Group Description:

5. Select the reports to include in this group and click Add.



Available Reports

- DPA-SUSE-MYSQL56:3306 - Top SQL
- DPA-SUSE-MYSQL56:3306 - Top Waits
- DPAORA11PER\_DPAORA11PER - Top
- DPAORA11PER\_DPAORA11PER - Top
- DPASQL2K12 - Top SQL
- DPASQL2K12 - Top Waits
- DPASQL2K14-BI - Top SQL
- DPASQL2K14-BI - Top Waits
- DPASYB157DB2105:50000 - Top SQL
- DPASYB157DB2105:50000 - Top Waits

Group Reports

- Waits for MV\_REFRESH INSERT
- Waits for SELECT FROM CON\_METRI

Add

Remove

6. Click OK.

This group is added to the list of report groups.



## Link together separate DPA servers

Use Central Server mode to link separate DPA servers together. This is useful if you want to monitor more than 250 database instances, or if your monitored databases are distributed geographically. See the following topics:

- [Set up a Central Server and add remote DPA servers](#)
- [Configure authentication for the DPA Central Server](#)
- [View information from remote servers on the DPA Central Server](#)
- [Advanced configuration for the DPA Central Server](#)

## Set up a Central Server and add remote DPA servers

Use Central Server mode to link separate DPA servers together. This is useful if:

- You want to monitor more than 250 database instances. You can divide monitoring tasks between different DPA servers.
- Your monitored databases are distributed geographically. You can install separate DPA servers in each location.

The Central Server collects information from your remote servers and consolidates the data into a single interface. The Central Server has low overhead and no additional information is added to its repository database.

### Set up a Central Server


1. Install DPA on a server. This will be your Central Server.
2. Log in to that instance as an administrator.
3. On the DPA menu, click Options.
4. Under Administration > Display, click Manage Central.

Your DPA server should be listed as the Central DPA Server in the list of Registered Servers.

### Add remote DPA servers

The user credentials for the Central and remote DPA servers must match. See [Configure authentication for the DPA Central Server](#) for more information.

1. On the DPA menu, click Options.
2. Under Administration > Display, click Manage Central.
3. Click Add Server.
4. Enter information about the remote DPA server.
5. Click Test connection, and click Save.

 A successful test indicates that DPA can communicate with the remote server through the provider host and port. It does not indicate that DPA can authenticate users.

If the test fails, check the host name in the Server Name field. Does it contain an underscore ( `_` ) character? An underscore is not valid for host names. If you cannot rename the host, enter the IP address.

6. Repeat steps 1 - 4 for the remaining remote DPA servers.

The details of your remote DPA servers are not stored in the repository, but in a file on your Central Server, located here:

```
<DPA_Home>/iwc/tomcat/ignite_config/iwc/central/RemoteRepositories.json
```


This is a plain-text JavaScript Object Notation (JSON) file. No sensitive data is stored in this file.

## Configure authentication for the DPA Central Server

You can authenticate to the [Central Server](#) and the remote servers using one account. The account must be added to each server as a DPA user, or through an Active Directory (AD) or LDAP group.

### Log in with a DPA user

You must create the user on the Central Server and each remote server. See [Create a DPA user account and assign privileges](#) for more information.

 The password must match on all servers.

Read-only permissions are sufficient to view data from the remote repositories.

### Log in with an Active Directory or LDAP user

You must first set up AD or LDAP on the Central Server and each remote server. See [DPA user authentication options](#) for more information.

Next, create the AD or LDAP group of the user on the Central Server and each remote server. See [Create and manage DPA contacts and contact groups](#) for more information.

Read-only permissions are sufficient to view data from the remote repositories.

## View information from remote servers on the DPA Central Server

The default homepage of a [Central Server](#) is the DPA homepage. Navigate to the Central Server page to see database information from all registered remote servers.

1. Log in to your DPA Central Server as an administrator.
2. In the menu, click Central.

## Advanced configuration for the DPA Central Server

You may need to change the [Central Server](#) configuration to make it run more efficiently in your environment.

To change the default behavior, you can edit the `system.properties` file in the `/iwc/tomcat/ignite_config/idc` directory of your Central Server and add the desired setting.

### General Central Server settings

Setting	Value	Description
<code>com.confio.iwc.central.enabled</code>	true (default)  false	Enables or disables the use of Central Server mode.
<code>com.confio.iwc.token.login.supported</code>	true (default)  false	Enables or disables the use of encrypted login tokens when jumping from the Central Server to a remote instance.  If true, a web service call authenticates the user and creates a temporary token to identify the incoming user and bypass the login process.  If false, the user is always prompted to log in to the remote instance.

Setting	Value	Description
com.confio.iwc.show.all.errors	true false (default)	Determines which users see failures in the Unavailable DPA Servers section. If true, all users see failures for all instances. If false, only administrators see failures. Set this option to false if you do not want all users to know about other DPA instances in the organization.
com.confio.iwc.automatic.update	true (default) false	Enables or disables a process that performs simple checks on the file when DPA starts. For example, flagging any local instances as the Central Server.
com.confio.iwc.alarm.level	Warning	The minimum message level to include on the Alarm Details tab. Valid values are below. If (empty) is set, details are disabled. <ul style="list-style-type: none"> <li>• Critical</li> <li>• Warning</li> <li>• Normal</li> <li>• (empty)</li> </ul>
com.confio.iwc.alarm.count	200	The number of detail rows to show on the Alarm Details tab.

## Thread pool settings

These settings control the number of threads that are used by the Central Server to make web service calls to other remote servers. The default settings are set for a few concurrent users hitting up to 100 remote instances. If you have more than 100 instances or many concurrent users, SolarWinds recommends adjusting these settings higher.

Setting	Value	Description
com.confio.iwc.centralServiceTaskExecutor.corePoolSize	20	The core number of threads that Central Server uses to make web service calls to remote servers.

Setting	Value	Description
com.confio.iwc.centralServiceTaskExecutor.maxPoolSize	40	The maximum number of threads that Central Server uses to make web service calls to the remote servers. Central Server adds more threads only when all core threads are in use and the task queue is full.
com.confio.iwc.centralServiceTaskExecutor.queueCapacity	1000	The maximum number of requests in the queue before Central Server either creates new threads to help with the work or rejects the request. Tasks are rejected if all 40 threads cannot keep up with the requests being made.
com.confio.iwc.centralServiceTaskExecutor.keepAliveSeconds	120	The number of seconds to keep an idle thread before removing it.

## Client factory cache

A client factory creates web service clients that talk to remote instances on a per-user basis. One client factory is created per host:port combination (not per user), so the same factory is used to create individual clients for different users. Factory creation is resource-intensive because an initial handshake is done between the client and server, and kept in a cache for reuse.

Setting	Value	Description
com.confio.iwc.client.factory.cache.size	100	<p>The maximum number of client factories held in the cache.</p> <p>The default is 100, which equates to 100 unique remote DPA instances.</p> <p>Increase this value if you are connecting to more than 100 remote instances.</p>
com.confio.iwc.client.factory.cache.timeout	1800	<p>The number of seconds a client factory remains in the cache without being used.</p> <p>The default is 1800 seconds, which is equal to 30 minutes.</p>
com.confio.iwc.client.factory.connection.timeout	15	<p>The number of seconds a client attempts to establish a connection before it times out.</p> <p>The default is 15.</p> <p>Zero (0) specifies that the client will continue to attempt to open a connection indefinitely.</p>
com.confio.iwc.client.factory.read.timeout	30	<p>The number of seconds the client waits for a response before it times out.</p> <p>The default is 30 seconds.</p> <p>Zero (0) specifies that the client will wait indefinitely.</p>
com.confio.iwc.client.factory.enable.chunking	true false (default)	<p>Enables or disables HTTP chunking.</p> <p>False is the safer option.</p>

Setting	Value	Description
com.confio.iwc.client.factory.enable.log	true (default)  false	Enables logging of inbound and outbound messaging to capture the web service calls. Log levels are still controlled in the <code>log4j.xml</code> file.  Set this value to false to disable logging.

## Use the DPA REST API

Use the DPA REST API to securely connect to the DPA server and issue commands. DPA API calls can retrieve information and automate management tasks, such as registering database instances, stopping and starting monitors, adding annotations, and allocating licenses.

See the following topics to learn more about using the DPA API:

- [Manage tokens used for authentication to the DPA API](#)
- [Learn about and experiment with the DPA API](#)
- [Examples of using Python scripts to make DPA API calls](#)
- [Examples of PowerShell scripts that make DPA API calls](#)

## Manage tokens used for authentication to the DPA API

Two types of tokens are required to authenticate requests to the DPA API:

- An **access token** is required to make authenticated calls to the DPA API. Access tokens are obtained when needed (for example, [at the beginning of a script](#) that makes API calls, or when you use the Swagger interface to [experiment with the DPA API](#)).

By default, access tokens expire after 900 seconds. You can change the default through the [advanced option](#) `API_ACCESS_TOKEN_EXPIRATION`. Access tokens also expire if the DPA server is rebooted.

- A **refresh token** is used to obtain access tokens. Refresh tokens are obtained through the DPA interface by an administrator and stored in a secure location, as described below.

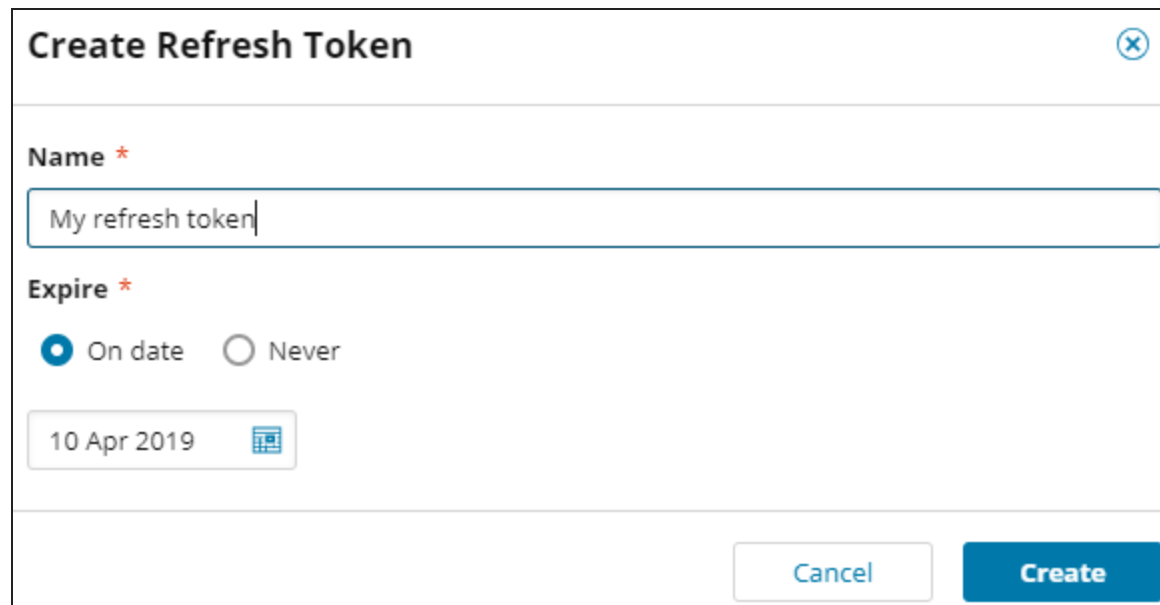
Refresh tokens typically have long lifespans. When you create a refresh token, you can specify the expiration date or set it to never expire. The default expiration date is after 90 days. You can change the default through the [advanced option](#) `API_REFRESH_TOKEN_EXPIRATION`.

## Create a refresh token

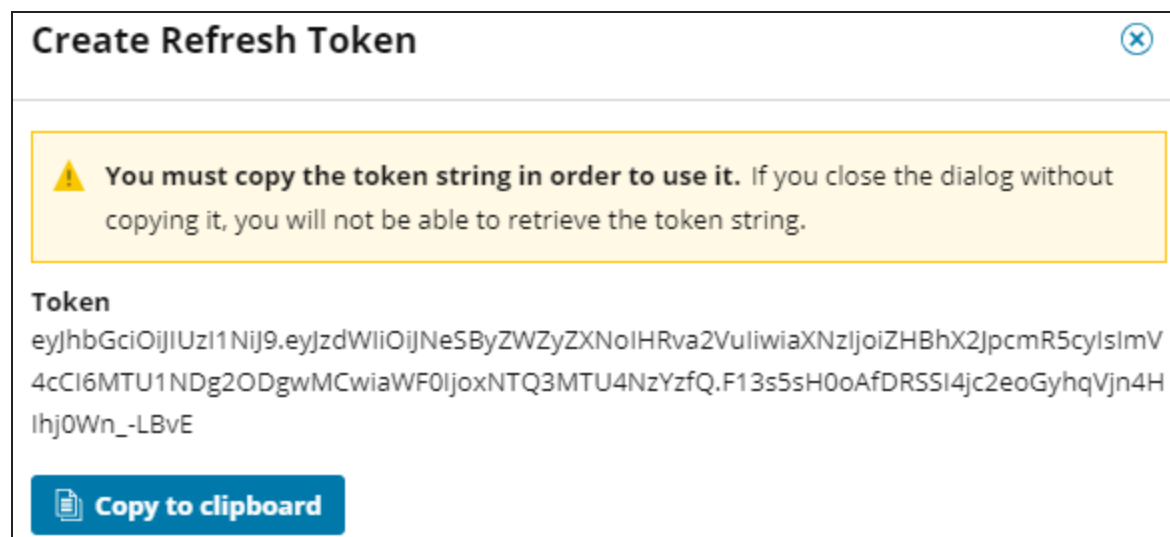
1. Log in to DPA as a user with administrative privileges.
2. On the DPA menu, click Options.
3. Under Users & Contacts, click Refresh Token Management.
4. On the API Refresh Token Management page, click Create token.
5. Enter a name and specify when the token expires.




By default, refresh tokens for the DPA API expire after 90 days. However, you can choose to create refresh tokens that never expire.



6. Click Create. The token string is displayed.



7. Click Copy to clipboard, and then click Close.

 If you create a refresh token and fail to copy the string or lose the copied string, the refresh token cannot be used. Delete that token and create a new one.

## About storing refresh tokens

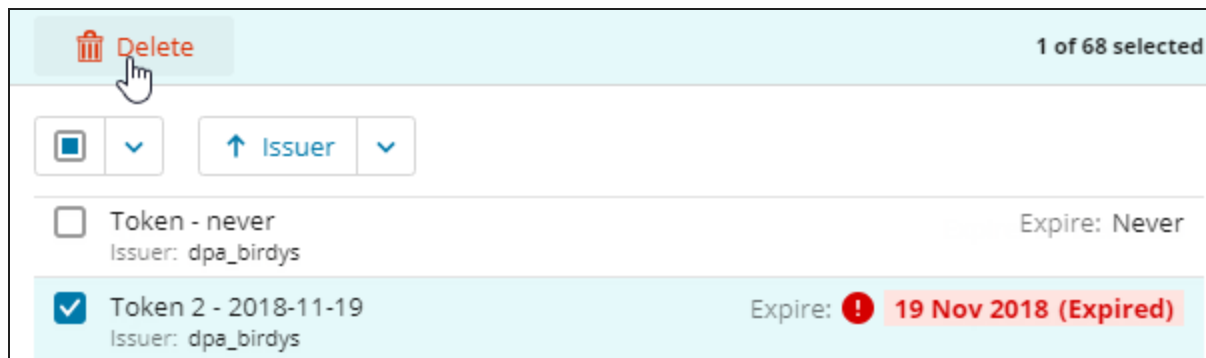
Store refresh tokens in a secure location, such as a password-protected file system or an encrypted database. Limit access to users who need the tokens to make API calls.

If you believe that a refresh token has been accessed by an unauthorized user, delete it and create a new one.

## Delete a refresh token

You can delete a refresh token at any time. For example, you can delete refresh tokens that have expired. If you delete a refresh token that has not expired, any access tokens obtained using that refresh token are invalidated and can no longer be used.

1. Log in to DPA as a user with administrative privileges.
2. On the DPA menu, click Options.
3. Under Users & Contacts, click Refresh Token Management.
4. On the API Refresh Token Management page, select one or more tokens.
5. Click Delete.



## Learn about and experiment with the DPA API

The DPA API is documented in the Swagger interface. Use this interactive interface to explore the available API endpoints and try out API calls.

 You can also see examples of [Python](#) and [PowerShell scripts](#) that call the DPA API.

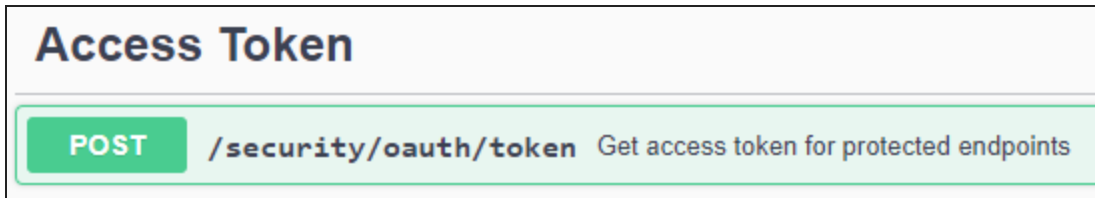
## Access the DPA API documentation and get authorization to make API calls

You can access the URL and review the DPA API documentation without being authorized to make API calls. However, an [access token](#) is required to make API calls. Complete the following steps to access the API documentation and authenticate with an access token.

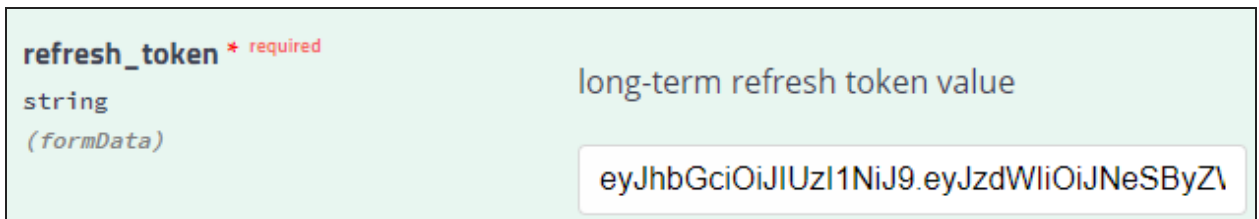
1. [Create a new refresh token](#) and copy it to the clipboard, or copy an existing refresh token that your organization has stored in a secure location.
2. On the DPA menu, click Options.
3. Under Support > Utilities, click Management API Documentation.

The Swagger interface opens.

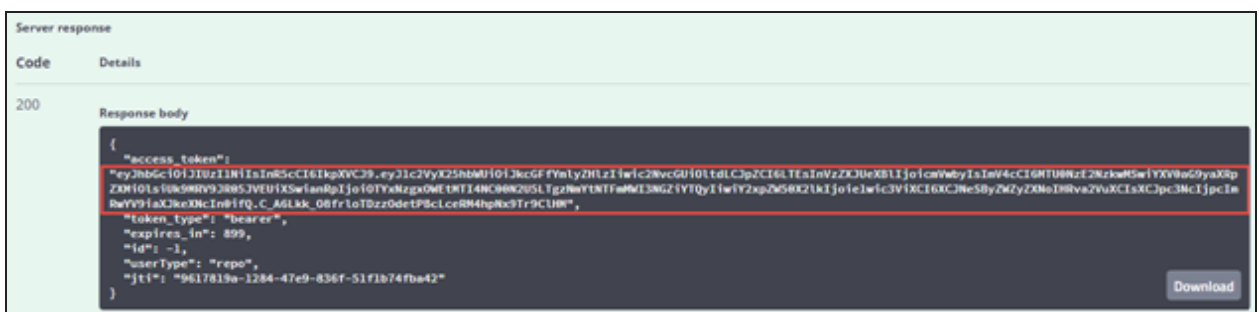
4. Use the refresh token to obtain an access token:
  - a. Click Access Token to expand it.



- b. Click Post to expand that section.
- c. Click the Try it out button.
- d. Paste the refresh token value you copied in step 1 into the refresh\_token box.

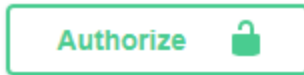


- e. Click Execute.
- f. Copy the access token from the Response body.



5. Authenticate with the access token:

- a. In the upper-right corner, click Authorize.



- b. In the Available authorizations dialog, type `bearer` followed by a space, and then paste the access token.

### Available authorizations

---

**Bearer (apiKey)**

Name: Authorization  
In: header  
Value:

- c. Click Authorize.

If the authorization is successful, the following dialog is displayed.

### Available authorizations

---

**Bearer (apiKey)**

**Authorized**

Name: Authorization  
In: header  
Value: \*\*\*\*\*

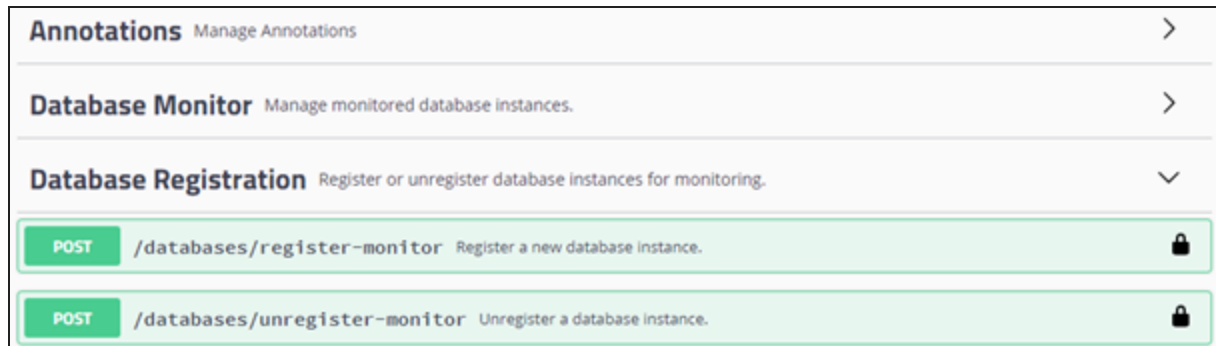
- d. Click Close.

You can now use the Swagger interface to learn about and execute the available API commands.

## View the DPA API documentation

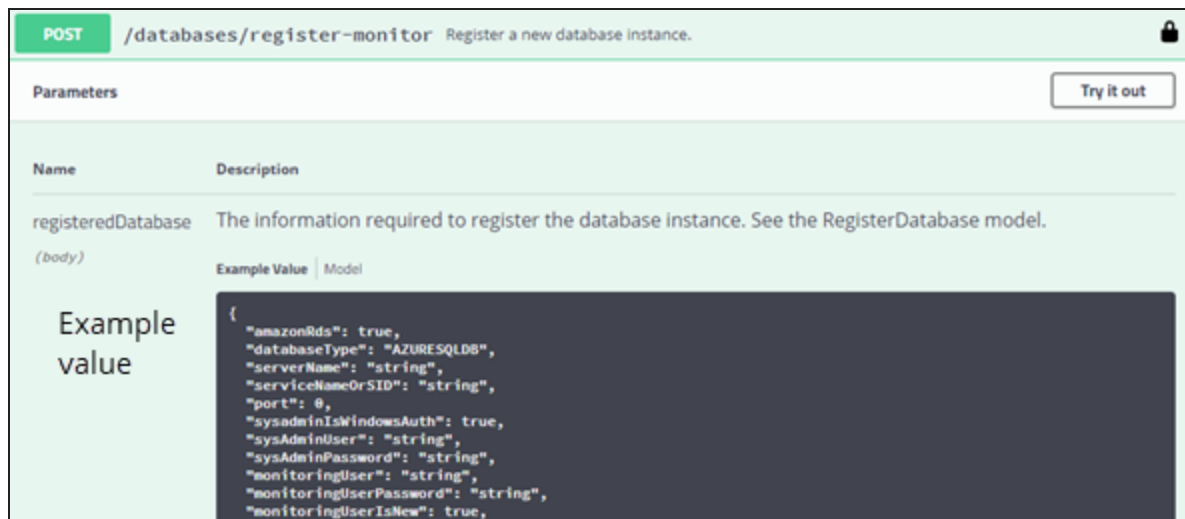
The Management API spec provides detailed information about each API endpoint. Endpoints are grouped by function.

1. Click any group to display the endpoints within it.

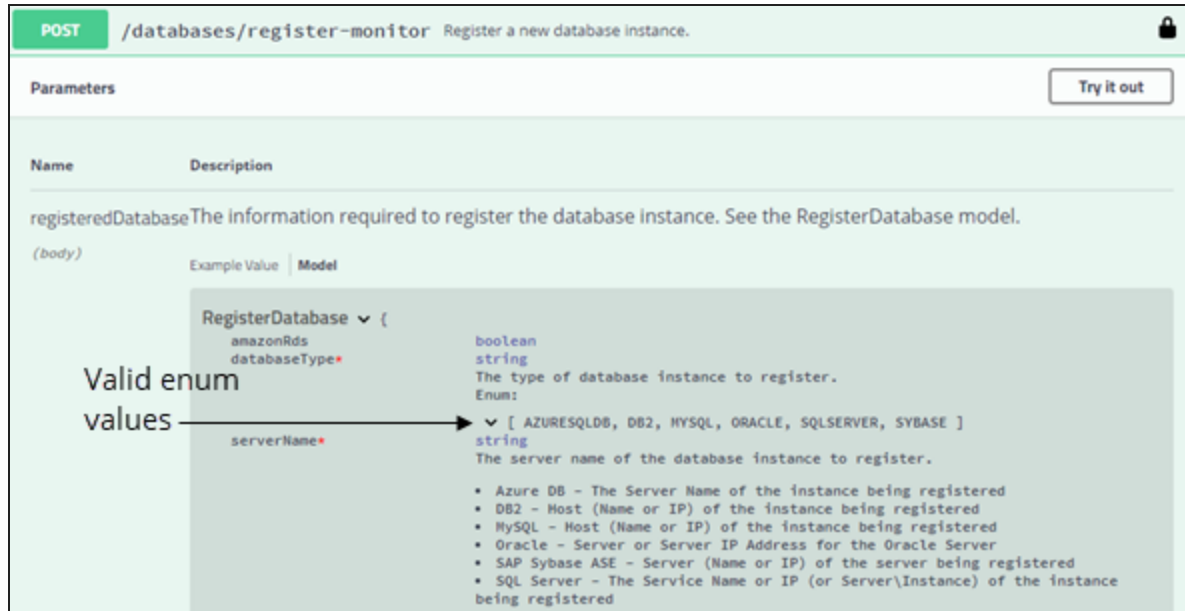


2. Click the endpoint to display its parameters and responses.

Complex parameters and responses include an Example Value | Model section. The example value is shown by default.



- Click Model to display additional information, including the valid values for enumerations.



POST /databases/register-monitor Register a new database instance.

Parameters Try it out

Name	Description
registeredDatabase	The information required to register the database instance. See the RegisterDatabase model.

(body)

Example Value | Model

```

RegisterDatabase {
  amazonRds boolean
  databaseType* string
  Enum:
  [ AZURESQLDB, DB2, MYSQL, ORACLE, SQLSERVER, SYBASE ]
  serverName* string
  The server name of the database instance to register.
  • Azure DB - The Server Name of the instance being registered
  • DB2 - Host (Name or IP) of the instance being registered
  • MySQL - Host (Name or IP) of the instance being registered
  • Oracle - Server or Server IP Address for the Oracle Server
  • SAP Sybase ASE - Server (Name or IP) of the server being registered
  • SQL Server - The Service Name or IP (or Server\Instance) of the instance being registered

```

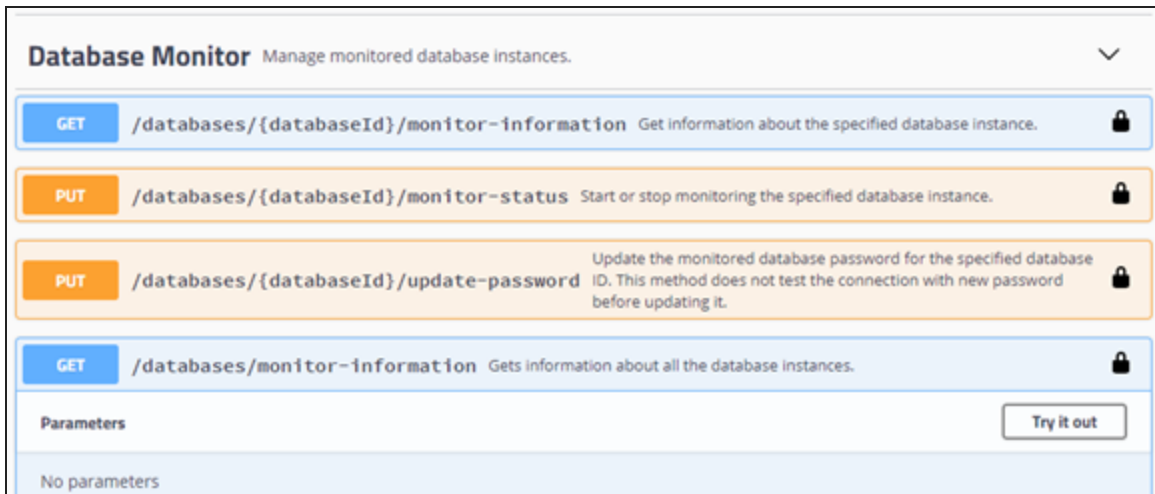
Valid enum values →

## Make an API call from the Swagger interface

The following example shows how to make a call to get the current license allocation for a monitored database instance.

**i** When you make an API call through the Swagger interface, the call affects your DPA server in the same way as it would if it were issued through a command or script.

1. If you do not know the database ID, complete the following steps to get it:
  - a. Click Database Monitor to display the endpoints.
  - b. Click GET/databases/monitor-information to expand it.

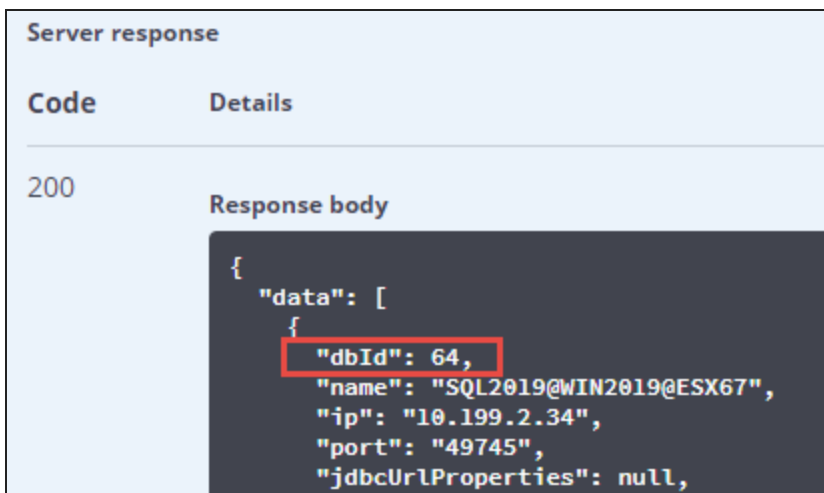


The screenshot shows the 'Database Monitor' section of an API management console. It lists several endpoints for managing database instances. The first endpoint is highlighted in blue:

Method	Endpoint	Description	Lock
GET	/databases/{databaseId}/monitor-information	Get information about the specified database instance.	🔒
PUT	/databases/{databaseId}/monitor-status	Start or stop monitoring the specified database instance.	🔒
PUT	/databases/{databaseId}/update-password	Update the monitored database password for the specified database ID. This method does not test the connection with new password before updating it.	🔒
GET	/databases/monitor-information	Gets information about all the database instances.	🔒

Below the endpoints, there is a 'Parameters' section with a 'Try it out' button and the text 'No parameters'.

- c. Click Try it out, and then click Execute.
    - d. Scroll through the Response body, find the database name, and make a note of the associated ID.

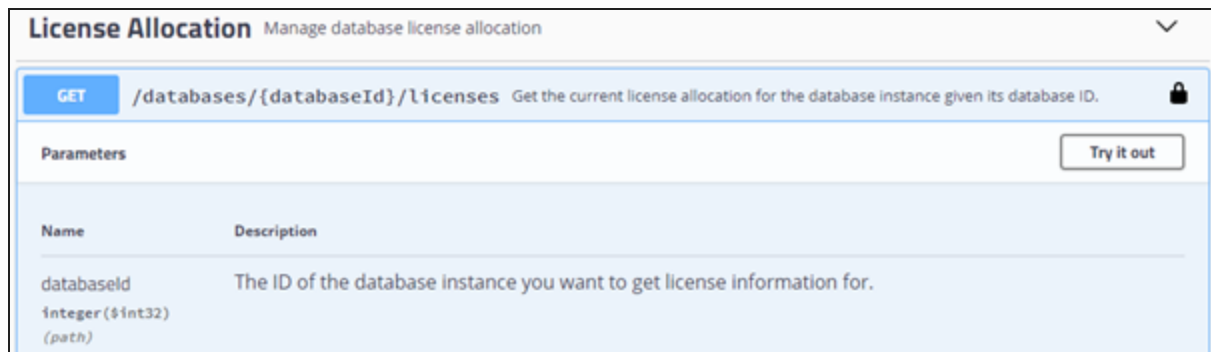


The screenshot shows the 'Server response' for the GET/databases/monitor-information endpoint. The response code is 200. The response body is a JSON array containing one object with the following properties:

```
{
  "data": [
    {
      "dbId": 64,
      "name": "SQL2019@WIN2019@ESX67",
      "ip": "10.199.2.34",
      "port": "49745",
      "jdbcUrlProperties": null,
    }
  ]
}
```

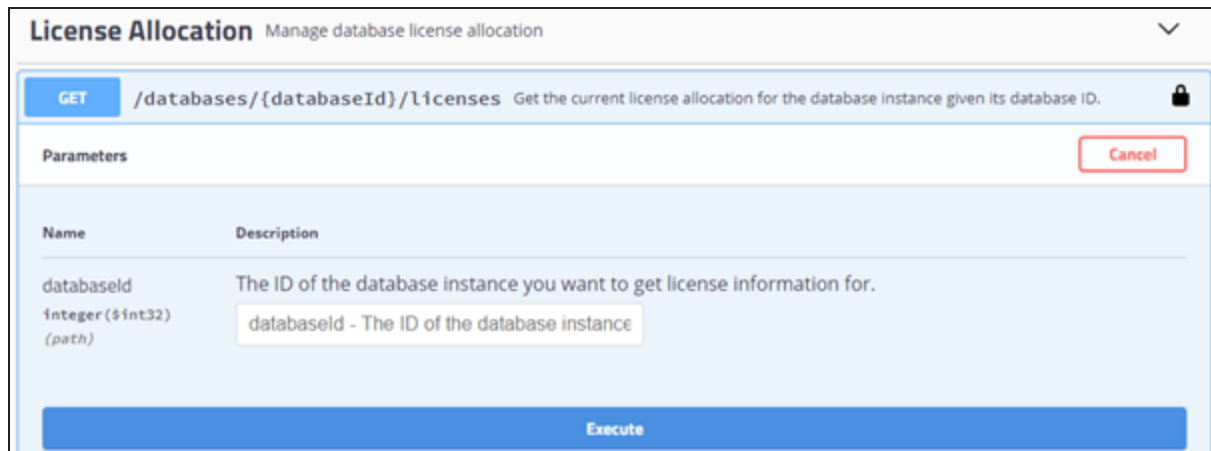
The 'dbId' value '64' is highlighted with a red box.

2. Click License Allocation to display the endpoints.
3. Click the GET/databases/{databaseId}/licenses endpoint to expand it.



4. Click Try it out.

The interface displays a field for the parameter value and an Execute button.



5. Enter the database ID and click Execute.

The Response body section shows the response from the DPA server, and the Curl section shows the Curl command (including the access token) that could be run to make this API call.





- [License Allocation examples](#)
- [Annotation examples](#)
- [Database Registration examples](#)
- [Database Custom Properties examples](#)
- [Full working script](#)

## Prerequisites

- Before you can use scripts to make API calls, you must [create a refresh token](#).
- These examples use the Requests HTTP library for Python. This library must be installed for these examples to work.

## If your DPA server does not use HTTPS or your certificates are self-signed

The examples all use HTTPS, which can cause problems if your DPA server is not configured to use HTTPS or if your certificates are self signed. If this is the case, you can do either of the following:

- Run the examples using HTTP.
- Change the `verify_cert` value to `False` in the configuration section to prevent verifying the server's TLS certificate.

```
# =====  
# Configure the variables below for the DPA Host  
# =====  
base_url = "https://localhost:8124/iwc/api/"  
refresh_token = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."  
verify_cert = False  
# =====
```

## Get an access token

The first step in using the API is to get an access token. An access token is required to make any API calls. This call POSTs the [refresh token](#) to DPA, which returns an access token to be used by all other API calls.

- If the call is successful, it prints out the data that was returned from DPA, including the `access_token`, and then goes on to create HTTP Headers that will contain the access token and other information to be used on subsequent calls.
- If the call is not successful it prints out the error message.

You must set the `base_url` and the `refresh_token` variables to match your environment.

```
# =====
# Configure the variables below for the DPA Host
# =====
base_url = "https://localhost:8124/iwc/api/"
refresh_token = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."
verify_cert = True
# =====

# =====
# Get Access Token
# =====

def get_access_header(prefix_url, rfrsh_token):
    """
    Given a base url and a refresh token retrieve the access token
    and return a header object with it.
    :param prefix_url: the base url
    :param rfrsh_token: refresh token used to get access token
    :return: the request header that contains the access token
    :rtype: dict
    """

    auth_token_url = prefix_url + "security/oauth/token"
    grant_type = "refresh_token"

    payload = {"grant_type": grant_type, "refresh_token": rfrsh_token}
    try:
        # get an access token
        resp = requests.post(auth_token_url, data=payload, verify=verify_cert)
        resp.raise_for_status()
        resp_json = resp.json()

        token_type = resp_json["token_type"]
        access_code = resp_json["access_token"]

        headers = {"authorization": f"{token_type} {access_code}",
                  "content-type": "application/json;charset=UTF-8",
                  "accept": "application/json"}
    }

    return headers
```

```
except requests.exceptions.HTTPError as ex:
    print(ex)
    print(ex.response.text)
    # print(json.dumps(json.loads(ex.response.text), indent=2))
    return None # requests is bad return None, can't get access_code

# get the header that contains access token for authentication
header = get_access_header(base_url, refresh_token)
if header is None:
    sys.exit(0)
```

## Database Monitor examples

The following examples show how to use Database Monitor calls.

### Get information about one monitored database instance

```
# Get information about a single monitored database instance
database_id = 1
monitor_url = f"{base_url}databases/{database_id}/monitor-information"
single_monitor = None
try:
    print(f"\n*** Get Monitor Information for database with id of {database_id} ***")
    response = requests.get(monitor_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    single_monitor = response_json["data"]
    print(json.dumps(single_monitor, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Get Monitor Information for database with id of 1 ***
{
  "dbId": 1,
  "name": "DEV-DPA\\SQLEXPRESS",
  "ip": "127.0.0.1",
  "port": "1433",
  "jdbcUrlProperties": "applicationIntent=readOnly",
  "connectionProperties": null,
```

```
"databaseType": "SQL Server",
"databaseVersion": "12.0.6205.1",
"databaseEdition": "Enterprise Edition: Core-based Licensing (64-bit)",
"monitoringUser": "ignite_next",
"defaultDbLicenseCategory": "DPACAT2",
"assignedDbLicenseCategory": "DPACAT2",
"assignedVmLicenseCategory": null,
"monitorState": "Monitor Stopped",
"oldestMonitoringDate": "2018-12-09T00:00:00.000-07:00",
"latestMonitoringDate": "2019-01-07T00:00:00.000-07:00",
"agListenerName": null,
"agClusterName": null,
"agName": null,
"racInfo": null,
"rac": false,
"rds": false,
"ebusiness": false,
"linkedToVirtualMachine": false,
"pdb": false
}
```

## Start and stop monitoring a database instance given its database ID

```
database_id = 1
monitor_url = f"{base_url}databases/{database_id}/monitor-status"
try:
    # Start monitoring a database instance given its database ID.
    print(f"*** Start Monitor for database {database_id} ***")
    body = {"command": "START"}
    response = requests.put(monitor_url, json=body, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))

    print("Waiting 15 seconds...")
    time.sleep(15)

    # Stop monitoring a database instance given its database ID.
    print(f"*** Stop Monitor for database {database_id} ***")
    body = {"command": "STOP"}
    response = requests.put(monitor_url, json=body, headers=header, verify=verify_cert)
```

```
response.raise_for_status()
response_json = response.json()
print(json.dumps(response_json["data"], indent=2))
print("Waiting 15 seconds...")
time.sleep(15)

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Start Monitor for database 1 ***
"SUCCESS"
Waiting 15 seconds...

*** Stop Monitor for database 1 ***
"SUCCESS"
Waiting 15 seconds...
```

## Get information about all monitored database instances

```
database_id = 1
monitor_url = f"{base_url}databases/monitor-information"
try:
    print("*** Get Information for a all database instances ***")
    response = requests.get(monitor_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
    print(json.dumps(data, indent=2))

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Get information for all database instances ***
[
  {
    "dbId": 1,
    "name": "DEV-DPA\\SQLEXPRESS",
```

```
"ip": "127.0.0.1",
"port": "1433",
"jdbcUrlProperties": "applicationIntent=readOnly",
"connectionProperties": null,
"databaseType": "SQL Server",
"databaseVersion": "12.0.6205.1",
"databaseEdition": "Enterprise Edition: Core-based Licensing (64-bit)",
"monitoringUser": "ignite_next",
"defaultDbLicenseCategory": "DPACAT2",
"assignedDbLicenseCategory": "DPACAT2",
"assignedVmLicenseCategory": null,
"monitorState": "Monitor Stopped",
"oldestMonitoringDate": "2018-12-09T00:00:00.000-07:00",
"latestMonitoringDate": "2019-01-07T00:00:00.000-07:00",
"agListenerName": null,
"agClusterName": null,
"agName": null,
"racInfo": null,
"rac": false,
"rds": false,
"ebusiness": false,
"linkedToVirtualMachine": false,
"pdb": false
},
{
  "dbId": 2,
  "name": "DEV-MYSQL",
  "ip": "127.0.0.1",
  ...
}
]
```

## Stop and start monitoring for all database instances

```
# Start monitoring all database instances.
monitor_url = f"{base_url}databases/monitor-status"
try:
    print("*** Starting all Monitors ***")
    body = {"command": "START"}
    response = requests.put(monitor_url, json=body, headers=header, verify=verify_cert)
    response.raise_for_status()
```

```
response_json = response.json()
print(json.dumps(response_json["data"], indent=2))
print("Waiting 30 seconds...")
time.sleep(30)

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Stop monitoring all database instances.
try:
    print("*** Stopping all Monitors ***")
    body = {"command": "STOP"}
    response = requests.put(monitor_url, json=body, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))
    print("Waiting 30 seconds...")
    time.sleep(30)

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Starting all Monitors ***
"SUCCESS"
Waiting 30 seconds...

*** Stopping all Monitors ***
"SUCCESS"
Waiting 30 seconds...
```

## Update the user password for a monitored database instance

```
database_id = 1
monitor_url = f"{base_url}databases/{database_id}/update-password"
try:
    print(f"*** Update the Monitor password for database {database_id} ***")
    body = {"password": "NewPassword!"}
    response = requests.put(monitor_url, json=body, headers=header, verify=verify_cert)
```



```
response.raise_for_status()
response_json = response.json()
print(json.dumps(response_json["data"], indent=2))

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Update the Monitor password for database 1 ***
"SUCCESS"
```

## License Allocation examples

The examples below show how to use License Allocation calls.

### Get information about currently installed licenses

```
license_url = f"{base_url}databases/licenses/installed"
try:
    print("\n*** Getting Installed license information with total amounts available
        for use and total amounts used ***")
    response = requests.get(license_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
    print("licenseProduct licenseCategory licensesAvailable licensesConsumed")
    print("-----")
    for i in range(len(data)):
        print('{:<15s}{:<16s}{:>17d}{:>17d}'.format(data[i]["licenseProduct"], data[i]
            ["licenseCategory"], data[i]["licensesAvailable"], data[i]["licensesConsumed"]))

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
```

```

*** Getting Installed license information with total amounts available
    for use and total amounts used ***
licenseProduct licenseCategory licensesAvailable licensesConsumed
-----
DPACAT1        DPA_DB          100             22
DPACAT2        DPA_DB          100             16
DPAAzureSQL    DPA_DB           0               0
DPAVM          DPA_VM          100             12

```

## Get license information for a single database instance

```

database_id = 1
license_url = f"{base_url}databases/{database_id}/licenses"
try:
    print(f"\n*** Getting current license information for the database instance with
database ID of {database_id} ***")
    response = requests.get(license_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Getting current license information for the database instance with database ID of 1
***
{
  "serverName": "DEV-DPA",
  "overLicensed": false,
  "performanceLicenseProduct": "DPACAT2",
  "vmLicenseProduct": "DPAVM"
}

```

## Update license information for a database instance

```

database_id = 1
license_url = f"{base_url}databases/{database_id}/licenses"

```

```
# Add a DPACAT2 and a DPAVM license
body = {"performanceLicenseProduct": "DPACAT2",
        "vmLicenseProduct": "DPAVM"}
try:
    print(f"\n*** Updating license for database id {database_id} ***")
    response = requests.put(license_url, json=body, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Remove the DPAVM license
body = {"performanceLicenseProduct": "DPACAT2",
        "vmLicenseProduct": "REMOVE"}
try:
    print(f"\n*** Updating license for database id {database_id} ***")
    response = requests.put(license_url, json=body, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Updating license for database id 1 ***
{
    "serverName": "DEV-BOU-CALLEN",
    "overLicensed": false,
    "performanceLicenseProduct": "DPACAT2",
    "vmLicenseProduct": DPAVM
}
*** Updating license for database id 1 ***
{
    "serverName": "DEV-BOU-CALLEN",
    "overLicensed": false,
    "performanceLicenseProduct": "DPACAT2",
    "vmLicenseProduct": null
}
```

```
}
```

## Annotation examples

The examples below show how to use Annotation calls.

### Get a list of annotations for the last 30 days

```
# Gets a List of annotations for the last 30 days
database_id = 1
annotation_url = f"{base_url}databases/{database_id}/annotations"

# Dates are in ISO 8601 format ( 2018-12-31T12:00:00.000-07:00 )
end_time = datetime.datetime.now()
start_time = end_time + datetime.timedelta(days=-30)
args = {"startTime": start_time.astimezone().isoformat(),
        "endTime": end_time.astimezone().isoformat()}

try:
    print("\n*** Getting Annotations for the last 30 days ***")
    response = requests.get(annotation_url, params=args, headers=header, verify=verify_
cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Getting Annotations for the last 30 days ***
[
  {
    "id": 112,
    "title": "Test Title API",
    "description": "Test Event created by DPA API",
    "createdBy": "DPA API",
    "time": "2018-12-11T10:01:35-07:00",
    "type": "API"
  },
  {
```

```
"id": 113,  
"title": "Test Title API",  
"description": "Test Event created by DPA API",  
"createdBy": "DPA API",  
"time": "2018-12-12T15:00:40-07:00",  
"type": "API"  
},  
{  
  ...  
}  
]
```

## Create a new annotation

```
database_id = 1  
annotation_url = f"{base_url}databases/{database_id}/annotations"  
  
# Dates are in ISO 8601 format, no millis ( 2018-12-31T12:00:00-07:00 )  
create_time = datetime.datetime.now().replace(microsecond=0)  
body = {"title": "API Test Title",  
        "description": "API Test Description",  
        "createdBy": "Test API User",  
        "time": create_time.astimezone().isoformat()}  
try:  
    print("\n*** Creating Annotation ***")  
    response = requests.post(annotation_url, json=body, headers=header, verify=verify_  
cert)  
    response.raise_for_status()  
    response_json = response.json()  
    print(json.dumps(response_json["data"], indent=2))  
except requests.exceptions.HTTPError as e:  
    print(e)  
    print(e.response.text)  
  
# This will print out data like this:  
*** Creating Annotation ***  
{  
  "id": 171,  
  "title": "API Test Title",  
  "description": "API Test Description",  
  "createdBy": "Test API User",
```

```
"time": "2019-01-09T11:04:33-07:00",
"type": "API"
}
```

## Delete an annotation

```
database_id = 1
annotation_id = 171
annotation_url = f"{base_url}databases/{database_id}/annotations/{annotation_id}"
try:
    print(f"\n*** Deleting Annotation with id of {annotation_id} ***")
    response = requests.delete(annotation_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    if response.status_code == 204:
        print(f"Annotation with id of {annotation_id} deleted")
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Deleting Annotation with id of 171 ***
Annotation with id of 171 deleted
```

## Database Registration examples

The examples below show how to use Database Registration calls.

### Register and unregister a SQL Server database instance for monitoring

This example registers a new SQL Server database instance, waits 60 seconds, and then unregisters the database instance.

```
# -----
# Register a SQL Server database instance for monitoring.
# -----
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "SQLSERVER",
        "serverName": "127.0.0.1",
        "port": "1433",
        "sysAdminUser": "sa",
        "sysAdminPassword": "Password",
```

```
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "displayName": "DPA_SQL2K12"}

new_db_id = None
try:
    print("\n*** Register SQL Server database ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    data = responseJson["data"]
    new_db_id = data["databaseId"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)

# -----
# Unregister the SQL Server database instance.
# -----
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "sa",
        "sysAdminPassword": "Password"}

try:
    print(f"\n*** Unregister SQL Server database [{new_db_id}] ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)
```

```
# This will print out data like this:
*** Register SQL Server database ***
{
  "databaseId": 77,
  "result": "SUCCESS"
}
Waiting 60 seconds...

*** Unregister SQL Server database [77] ***
{
  "databaseId": 77,
  "result": "SUCCESS"
}
```

## Register and unregister an Oracle database instance for monitoring

This example registers a new Oracle database instance, waits 60 seconds, and then unregisters the database instance.

```
# -----
# Register an Oracle database instance for monitoring.
# -----
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "ORACLE",
       "serverName": "127.0.0.1",
       "serviceNameOrSID": "DPA_ORA11R1",
       "port": "1521",
       "sysAdminUser": "system",
       "sysAdminPassword": "Password",
       "sysPassword": "Password",
       "monitoringUser": "dpa_test_m",
       "monitoringUserPassword": "Password",
       "monitoringUserIsNew": True,
       "monitoringUserTableSpace": "USERS",
       "monitoringUserTempTableSpace": "TEMP",
       "oracleEBusinessEnabled": False,
       "displayName": "DPA_ORA11R1"}

new_db_id = None
```



```
try:
    print("\n*** Register Oracle database ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    data = responseJson["data"]
    new_db_id = data["databaseId"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)

# -----
# Unregister the Oracle database instance.
# -----
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "system",
        "sysAdminPassword": "Password"}

try:
    print(f"\n*** Unregister Oracle database [{new_db_id}] ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Register Oracle database ***
{
  "databaseId": 78,
  "result": "SUCCESS"
```

```
}  
Waiting 60 seconds...  
  
*** Unregister Oracle database [78] ***  
{  
  "databaseId": 78,  
  "result": "SUCCESS"  
}
```

## Register and unregister a MySQL database instance for monitoring

This example registers a new MySQL database instance, waits 60 seconds, and then unregisters the database instance.

```
# -----  
# Register a MySQL database instance for monitoring.  
# -----  
registration_url = f"{base_url}databases/register-monitor"  
body = {"databaseType": "MYSQL",  
        "serverName": "127.0.0.1",  
        "port": "3306",  
        "sysAdminUser": "root",  
        "sysAdminPassword": "Password",  
        "monitoringUser": "dpa_test_m",  
        "monitoringUserPassword": "Password",  
        "monitoringUserIsNew": True,  
        "displayName": "DPA_MYSQL56"}  
  
new_db_id = None  
try:  
    print("\n*** Register MySQL database ***")  
    response = requests.post(registration_url, json=body, headers=header, verify=verify_  
cert)  
    response.raise_for_status()  
    responseJson = response.json()  
    data = responseJson["data"]  
    new_db_id = data["databaseId"]  
    print(json.dumps(data, indent=2))  
except requests.exceptions.HTTPError as e:  
    print(e)  
    print(e.response.text)
```

```
print("Waiting 60 seconds...")
time.sleep(60)

# -----
# Unregister the MySQL database instance.
# -----
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "root",
        "sysAdminPassword": "Password"}
try:
    print(f"\n*** Unregister MySQL database [{new_db_id}] ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will print out data like this:
*** Register MySQL database ***
{
  "databaseId": 79,
  "result": "SUCCESS"
}
Waiting 60 seconds...

*** Unregister MySQL database [79] ***
{
  "databaseId": 79,
  "result": "SUCCESS"
}
```

## Database Custom Properties examples

The examples below show how to use Database Custom Properties calls. Custom property values can be included in [custom email templates](#) for alert notifications.

### Create a custom property

This script creates a custom property and defines its name and description.

```
property_name = "Location"
property_description = "Location of the database server"
create_property_url = f"{base_url}databases/properties"
body = {
    "name": property_name,
    "description": property_description
}

property_id = None

try:
    print("\n*** Creating custom property ***")
    response = requests.post(create_property_url, json=body, headers=header,
verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
    property_id = data["id"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

#This will print out data like this:
{
    "id": 1,
    "name": "Location",
    "description": "Location of the database server"
}
```

### Create a custom property value

This script creates a value for the custom property created by the previous script.

```
property_id = 1
property_value = "New York"
create_value_url = f"{base_url}databases/properties/" + str(property_id) + "/values"
body = property_value

property_value_id = None

try:
    print("\n*** Creating custom property value ***")
    response = requests.post(create_value_url, data=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
    property_value_id = data["id"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

#This will print out data like this:
{
    "id": 1,
    "value": "New York"
}
```

## Assign a property value to a monitored database instance

This script assigns a property value to a monitored database instance.

```
property_id = 1
property_value_id = 1
database_id = 1
assign_property_value_url = f"{base_url}databases/" + str(database_id) + "/properties/"
+ str(property_id) + "/values/" + str(property_value_id)

try:
    print("\n*** Assigning custom property value ***")
    response = requests.post(assign_property_value_url, headers=header, verify=verify_
cert)
    response.raise_for_status()
```

```
if response.status_code == 200:
    print(f"Custom property value assigned to the DB with ID: {database_id}")
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

#This will print out data like this:
#Custom property value assigned to the DB with ID: 1
```

## Get all information about properties

This script returns information about all custom properties and their values.

```
get_properties_url = f"{base_url}databases/properties?require=assignment"

try:
    print("\n*** Getting custom property information ***")
    response = requests.get(get_properties_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

#This will print out data like this:
[
  {
    "id": 1,
    "name": "Location",
    "description": "Location of the database server",
    "values": [
      {
        "id": 1,
        "value": "New York",
        "assignment": [
          1
        ]
      }
    ]
  },
  "unassigned": [
```

```
    2
  ]
}
]
```

## Delete a custom property

This script deletes a custom property.

```
property_id = 1
delete_property_url = f"{base_url}databases/properties/" + str(property_id)

try:
    print("\n*** Deleting custom property ***")
    response = requests.delete(delete_property_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    if response.status_code == 204:
        print(f"Custom property with ID of {property_id} deleted")
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

#This will print out data like this:
#Custom property with ID of 1 deleted
```

## Full working script

The following script combines all of the examples shown above into a script that can be run.

```
import json
import sys
import time
import datetime
import requests

# =====
# Configure the variables below for the DPA Host
# =====

base_url = "http://localhost:8123/iwc/api/"
refresh_token = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."
verify_cert = True
```

```
# =====  
  
# =====  
# Get Access Token  
# =====  
def get_access_header(prefix_url, rfrsh_token):  
    """  
    Given a base url and a refresh token retrieve the access token  
    and return a header object with it.  
    :param prefix_url: the base url  
    :param rfrsh_token: refresh token used to get access token  
    :return: the request header that contains the access token  
    :rtype: dict  
    """  
  
    auth_token_url = prefix_url + "security/oauth/token"  
    grant_type = "refresh_token"  
  
    payload = {"grant_type": grant_type, "refresh_token": rfrsh_token}  
    try:  
        # get an access token  
        resp = requests.post(auth_token_url, data=payload, verify=verify_cert)  
        resp.raise_for_status()  
        resp_json = resp.json()  
  
        token_type = resp_json["token_type"]  
        access_code = resp_json["access_token"]  
  
        headers = {"authorization": f"{token_type} {access_code}",  
                  "content-type": "application/json;charset=UTF-8",  
                  "accept": "application/json"  
                }  
  
        return headers  
  
    except requests.exceptions.HTTPError as ex:  
        print(ex)  
        print(ex.response.text)  
        # print(json.dumps(json.loads(ex.response.text), indent=2))  
        return None # requests is bad return None, can't get access_code
```



```
# get the header that contains access token for authentication
header = get_access_header(base_url, refresh_token)
if header is None:
    sys.exit(0)

# =====
# Database Monitor Examples
# =====

# Calls for individual monitors...

# Get Monitor Information for a single database instance
database_id = 1
monitor_url = f"{base_url}databases/{database_id}/monitor-information"
single_monitor = None
try:
    print(f"\n*** Get Monitor Information for database with id of {database_id} ***")
    response = requests.get(monitor_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    single_monitor = response_json["data"]
    print(json.dumps(single_monitor, indent=2))

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Start or Stop monitoring a database instance given its database ID.
# If it is already running stop it and then restart it
# If it is not running start it and then stop it
if single_monitor is not None:
    monitor_url = f"{base_url}databases/{database_id}/monitor-status"
    if single_monitor["monitorState"] == "Monitor Running":
        change_command = "STOP"
        revert_command = "START"
    elif single_monitor["monitorState"] == "Monitor Stopped":
        change_command = "START"
        revert_command = "STOP"
    else:
        change_command = None
        revert_command = None
```

```
if change_command is not None:
    try:
        print(f"\n*** {change_command} Monitor for database {database_id} ***")
        body = {"command": change_command}
        response = requests.put(monitor_url, json=body, headers=header, verify=verify_
cert)
        response.raise_for_status()
        response_json = response.json()
        print(json.dumps(response_json["data"], indent=2))

        print("Waiting 15 seconds...")
        time.sleep(15)

        print(f"\n*** {revert_command} Monitor for database {database_id} ***")
        body = {"command": revert_command}
        response = requests.put(monitor_url, json=body, headers=header, verify=verify_
cert)
        response.raise_for_status()
        response_json = response.json()
        print(json.dumps(response_json["data"], indent=2))

        print("Waiting 15 seconds...")
        time.sleep(15)

    except requests.exceptions.HTTPError as e:
        print(e)
        print(e.response.text)

# Calls for all monitors...

# Get Monitor Information for all database instances
database_id = 1
running_ids = []
monitor_url = f"{base_url}databases/monitor-information"
try:
    print("\n*** Get Information for a all database instances ***")
    response = requests.get(monitor_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
```

```
print(json.dumps(data, indent=2))

# Keep a list of running or started monitors to be used later
for monitor in data:
    state = monitor["monitorState"]
    if state == "Monitor Running" or state == "Monitor Start No License" or 'Start' in
state:
        running_ids.append(monitor["dbId"])

print(f"Running Monitors: {running_ids}")

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Start monitoring all database instances.
monitor_url = f"{base_url}databases/monitor-status"
try:
    print("\n*** Starting all Monitors ***")
    body = {"command": "START"}
    response = requests.put(monitor_url, json=body, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))
    print("Waiting 30 seconds...")
    time.sleep(30)

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Stop monitoring all database instances.
try:
    print("\n*** Stopping all Monitors ***")
    body = {"command": "STOP"}
    response = requests.put(monitor_url, json=body, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))
    print("Waiting 30 seconds...")
    time.sleep(30)
```

```
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Try to put it back the way we found it by restarting the ones that were running
for db_id in running_ids:
    try:
        print(f"\n*** Starting Monitor for database {db_id} ***")
        monitor_url = f"{base_url}databases/{db_id}/monitor-status"
        body = {"command": "START"}
        response = requests.put(monitor_url, json=body, headers=header, verify=verify_cert)
        response.raise_for_status()
        response_json = response.json()
        print(json.dumps(response_json["data"], indent=2))
    except requests.exceptions.HTTPError as e:
        print(e)
        print(e.response.text)

# Update the monitor database user password (Un-comment to use)
#monitor_url = f"{base_url}databases/{database_id}/update-password"
#try:
#    print(f"*** Update the Monitor password for database {database_id} ***")
#    body = {"password": "Password"}
#    response = requests.put(monitor_url, json=body, headers=header, verify=verify_cert)
#    response.raise_for_status()
#    response_json = response.json()
#    print(json.dumps(response_json["data"], indent=2))

#except requests.exceptions.HTTPError as e:
#    print(e)
#    print(e.response.text)

# =====
# Licensing Examples
# =====

# Get the currently installed license information
license_url = f"{base_url}databases/licenses/installed"
try:
    print("\n*** Getting Installed license information with total amounts available for
```

```
use and total amounts used ***")
    response = requests.get(license_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
    print("licenseProduct licenseCategory licensesAvailable licensesConsumed")
    print("-----")
    for i in range(len(data)):
        print('{:<15s}{:<16s}{:>17d}{:>17d}'.format(data[i]["licenseProduct"], data[i]
["licenseCategory"],
        data[i]["licensesAvailable"], data[i]["licensesConsumed"]))

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Get License Information for a single database
license_url = f"{base_url}databases/{database_id}/licenses"
license_info = None
try:
    print(f"\n*** Getting current license information for the database instance with
database ID of {database_id} ***")
    response = requests.get(license_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    license_info = response_json["data"]
    print(json.dumps(license_info, indent=2))

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# This will Update License Information for a single database setting the
# Performance License and the VM License to what it currently is.
# It should succeed but it should make no changes.
if license_info is not None:
    database_id = 1
    license_url = f"{base_url}databases/{database_id}/licenses"
    db_product = license_info["performanceLicenseProduct"]
    vm_product = license_info["vmLicenseProduct"]
    body = {"performanceLicenseProduct": db_product,
```

```
    "vmLicenseProduct": vm_product}
try:
    print(f"\n*** Updating license for database id {database_id} ***")
    response = requests.put(license_url, json=body, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))

except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# =====
# Annotation Examples
# =====

# Gets a List of annotations for the last 30 days
annotation_url = f"{base_url}databases/{database_id}/annotations"

# Dates are in ISO 8601 format ( 2018-12-31T12:00:00.000-07:00 )
end_time = datetime.datetime.now()
start_time = end_time + datetime.timedelta(days=-30)
args = {"startTime": start_time.astimezone().isoformat(),
        "endTime": end_time.astimezone().isoformat()}

try:
    print("\n*** Getting Annotations for the last 30 days ***")
    response = requests.get(annotation_url, params=args, headers=header, verify=verify_
cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Create a new annotation
annotation_url = f"{base_url}databases/{database_id}/annotations"
annotation_id = None

#Dates are in ISO 8601 format, no millis ( 2018-12-31T12:00:00-07:00 )
```

```
create_time = datetime.datetime.now().replace(microsecond=0)
body = {"title": "API Test Title",
        "description": "API Test Description",
        "createdBy": "Test API User",
        "time": create_time.astimezone().isoformat()}
try:
    print("\n*** Creating Annotation ***")
    response = requests.post(annotation_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
    annotation_id = data["id"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Delete an annotation
if annotation_id is not None:
    annotation_url = f"{base_url}databases/{database_id}/annotations/{annotation_id}"
    try:
        print(f"\n*** Deleting Annotation with id of {annotation_id} ***")
        response = requests.delete(annotation_url, headers=header, verify=verify_cert)
        response.raise_for_status()
        if response.status_code == 204:
            print(f"Annotation with id of {annotation_id} deleted")
    except requests.exceptions.HTTPError as e:
        print(e)
        print(e.response.text)

# =====
# Registration Examples
# =====

# -----
# Register a SQL Server database instance for monitoring.
# -----
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "SQLSERVER",
        "serverName": "127.0.0.1",
```

```
        "port": "1433",
        "sysAdminUser": "User",
        "sysAdminPassword": "Password",
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "displayName": "DPA_SQL2K12"}

new_db_id = None
try:
    print("\n*** Register SQL Server database ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    data = responseJson["data"]
    new_db_id = data["databaseId"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)

# -----
# Un-register the SQL Server database instance.
# -----
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "User",
        "sysAdminPassword": "Password"}

try:
    print(f"\n*** Unregister SQL Server database [{new_db_id}] ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    print(json.dumps(responseJson["data"], indent=2))
```



```
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# -----
# Register an Oracle database instance for monitoring.
# -----
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "ORACLE",
        "serverName": "127.0.0.1",
        "serviceNameOrSID": "DPA_ORA11R1",
        "port": "1521",
        "sysAdminUser": "User",
        "sysAdminPassword": "Password",
        "sysPassword": "Password",
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "monitoringUserTableSpace": "USERS",
        "monitoringUserTempTableSpace": "TEMP",
        "oracleEBusinessEnabled": False,
        "displayName": "DPA_ORA11R1"}

new_db_id = None
try:
    print("\n*** Register Oracle database ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    data = responseJson["data"]
    new_db_id = data["databaseId"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)

# -----
```

```
# Un-register the Oracle database instance.
# -----
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "User",
        "sysAdminPassword": "Password"}
try:
    print(f"\n*** Unregister Oracle database [{new_db_id}] ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# -----
# Register a MySQL database instance for monitoring.
# -----
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "MYSQL",
        "serverName": "127.0.0.1",
        "port": "3306",
        "sysAdminUser": "User",
        "sysAdminPassword": "Password",
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "displayName": "DPA_MYSQL56"}

new_db_id = None
try:
    print("\n*** Register MySQL database ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    data = responseJson["data"]
```

```
new_db_id = data["databaseId"]
print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)
# -----
# Un-register the MySQL database instance.
# -----
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "User",
        "sysAdminPassword": "Password"}
try:
    print(f"\n*** Unregister MySQL database [{new_db_id}] ***")
    response = requests.post(registration_url, json=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    responseJson = response.json()
    print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)
# =====
# Custom Property Examples
# =====

# Create custom property
property_name = "Location"
property_description = "Location of the database server"
create_property_url = f"{base_url}databases/properties"
body = {
    "name": property_name,
    "description": property_description
}

property_id = None
```

```
try:
    print("\n*** Creating custom property ***")
    response = requests.post(create_property_url, json=body, headers=header,
verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
    property_id = data["id"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Create value of the custom property
property_value = "New York"
create_value_url = f"{base_url}databases/properties/" + str(property_id) + "/values"
body = property_value

property_value_id = None

try:
    print("\n*** Creating custom property value ***")
    response = requests.post(create_value_url, data=body, headers=header, verify=verify_
cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
    property_value_id = data["id"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Assign property value to DB
assign_property_value_url = f"{base_url}databases/" + str(database_id) + "/properties/"
+ str(property_id) + "/values/" + str(property_value_id)

try:
    print("\n*** Assigning custom property value ***")
    response = requests.post(assign_property_value_url, headers=header, verify=verify_
```

```
cert)
    response.raise_for_status()
    if response.status_code == 200:
        print(f"Custom property value assigned to the DB with ID: {database_id}")
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Get all information about properties (including DB assignment)
get_properties_url = f"{base_url}databases/properties?require=assignment"

try:
    print("\n*** Getting custom property information ***")
    response = requests.get(get_properties_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    data = response_json["data"]
    print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)


# Delete custom property
delete_property_url = f"{base_url}databases/properties/" + str(property_id)

try:
    print("\n*** Deleting custom property ***")
    response = requests.delete(delete_property_url, headers=header, verify=verify_cert)
    response.raise_for_status()
    if response.status_code == 204:
        print(f"Custom property with id of {property_id} deleted")
except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)
```

## Examples of PowerShell scripts that make DPA API calls

*The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.*

The following examples show PowerShell scripts that call the DPA API to retrieve information and perform DPA management functions. The first examples are snippets that demonstrate each API call individually. The last example is a full script that shows how to put the snippets together into a working script.

 You can call the DPA API with any programming language that can send HTTP requests. See [this topic](#) for Python script examples.

See the following sections:

- [Prerequisite](#)
- [If your DPA server does not use HTTPS or your certificates are self-signed](#)
- [Get an access token](#)
- [Database Monitor examples](#)
- [License Allocation examples](#)
- [Annotation examples](#)
- [Database Registration examples](#)
- [Database Custom Properties examples](#)
- [Full working script](#)

### Prerequisite

Before you can use scripts to make API calls, you must [create a refresh token](#).

## If your DPA server does not use HTTPS or your certificates are self-signed

The examples all use HTTPS, which can cause problems if your DPA server is not configured to use HTTPS or if your certificates are self signed. If this is the case, you can do either of the following:

- Run the examples using HTTP.
- Add the following code below the configuration section.

```
#-----  
# Adding certificate exception to prevent API errors  
#-----  
  
add-type @"  
    using System.Net;  
    using System.Security.Cryptography.X509Certificates;  
    public class TrustAllCertsPolicy : ICertificatePolicy {  
        public bool CheckValidationResult(  
            ServicePoint srvPoint, X509Certificate certificate,  
            WebRequest request, int certificateProblem) {  
            return true;  
        }  
    }  
"@  
[System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy
```

## Get an access token

The first step in using the API is to get an access token. An access token is required to make any API calls. This call POSTs the [refresh token](#) to DPA, which returns an access token to be used by all other API calls.

- If the call is successful, it prints out the data that was returned from DPA, including the `access_token`, and then goes on to create HTTP Headers that will contain the access token and other information to be used on subsequent calls.
- If the call is not successful it prints out the error message.

You must set the `$baseUrl` and the `$refreshToken` variables to match your environment.

```
#-----  
# Configure the variables below for the DPA Host  
#-----
```

```
$baseUrl = "https://localhost:8124/iwc/api/"
$refreshToken = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."

#-----
# Get an access token
#-----

$authTokenURL = $baseUrl + "security/oauth/token"
$body = @{"grant_type" = "refresh_token"
          "refresh_token" = "$refreshToken"}

Try {
    Write-Host "Getting Access Token..."
    $dpaAuthResponse = Invoke-RestMethod -Uri $authTokenURL -Method POST -Body $body
    $dpaAuthResponse | Format-List
}
Catch {
    $_.Exception.ToString()
    return
}

# If successful we will create our headers to be used for all API calls
$tokenType = $DpaAuthResponse.token_type
$accessToken = $DpaAuthResponse.access_token
$dpaHeader = @{}
$dpaHeader.Add("Accept", "application/json")
$dpaHeader.Add("Content-Type", "application/json;charset=UTF-8")
$dpaHeader.Add("Authorization", "$tokenType $accessToken")

# This will print out data like this:

Getting Access Token...

access_token :
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1IjoiYm9keiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."
otMSwidXNlcl...
token_type   : bearer
expires_in   : 365
id           : -1
userType     : repo
jti          : e0d51295-2010-4ed4-b5ea-982a4e6ae1c5
```



## Database Monitor examples

The following examples show how to use all Database Monitor calls.

### Get information about one monitored database instance

```
# Get Monitor Information for a single database
$databaseId = 1
$monitorURL = $baseURL + "databases/$databaseId/monitor-information"
Try {
    Write-Host "Get Monitor Information for database with id of $databaseId..."
    $monitorJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers $dpaHeader -
TimeoutSec 60
    $monitor = $monitorJSON.data
    $monitor | Format-List
}
Catch {
    $_.Exception.ToString()
}

# This will print out data like this:
Get Monitor Information for database with id of 1...
dbId                : 1
name                 : DEV-DPA\SQLEXPRESS
ip                   : 127.0.0.1
port                 : 1433
jdbcUrlProperties    : applicationIntent=readOnly
connectionProperties :
databaseType         : SQL Server
databaseVersion      : 12.0.6205.1
databaseEdition      : Enterprise Edition: Core-based Licensing (64-bit)
monitoringUser       : ignite_next
defaultDbLicenseCategory : DPACAT2
assignedDbLicenseCategory : DPACAT2
assignedVmLicenseCategory :
monitorState         : Monitor Running
oldestMonitoringDate : 2018-12-04T00:00:00.000-07:00
latestMonitoringDate : 2019-01-02T00:00:00.000-07:00
agListenerName      :
agClusterName       :
agName               :
racInfo              :
```

```
rac                : False
linkedToVirtualMachine : False
rds                : False
pdb                : False
ebusiness          : False
```

## Start and stop monitoring a database instance given its database ID

```
$databaseId = 1
$monitorURL = $baseURL + "databases/$databaseId/monitor-status"

# Start monitoring a database instance given its database ID.
Try {
    Write-Host "Start Monitor for database $databaseId..."
    $command = @{"command" = "START"} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
    $dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result"
    Write-Host "Waiting 15 seconds...`r`n"
    Start-Sleep -s 15
}
Catch {
    $_.Exception.ToString()
}

# Stop monitoring a database instance given its database ID.
Try {
    Write-Host "Stop Monitor for database $databaseId..."
    $command = @{"command" = "STOP"} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
    $dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result"
    Write-Host "Waiting 15 seconds...`r`n"
    Start-Sleep -s 15
}
Catch {
    $_.Exception.ToString()
}

# This will print out data like this:
```

```
Start Monitor for database 1...
Result: SUCCESS
Waiting 15 seconds...

Stop Monitor for database 1...
Result: SUCCESS
Waiting 15 seconds...
```

## Get information about all monitored database instances

```
# Get Monitor Information for all databases
$monitorURL = $baseURL + "databases/monitor-information"
Try {
    Write-Host "Get Monitor Information for all databases..."
    $monitorListJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers $dpaHeader
    -TimeoutSec 60
    $monitorList = $monitorListJSON.data
    $monitorList | Format-Table -AutoSize
}
Catch {
    $_.Exception.ToString()
}

# This will print out data like this:
Get Monitor Information for all databases...

dbId name                ip          port  jdbcUrlProperties
connectionProperties  databaseType databaseVersion ...
-----
-----
-----
1 DEV-DPA\SQLEXPRESS  10.10.10.1  1433  applicationIntent=readOnly
    SQL Server      12.0.6205.1  ...
3 DEVORA11_DEVORA11  10.10.10.2  1521
    Oracle          11.2.0.1.0   ...
10 DEV-MYSQL:3306    10.10.10.3  3306  dumpQueriesOnException=true
    MySQL          5.7.19       ...
etc.
```

## Stop and start monitoring for all database instances

```
$monitorURL = $baseURL + "databases/monitor-status"

# Start monitoring all database instances.
Try {
    Write-Host "Starting all Monitors..."
    $command = @{"command" = "START"} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
    $dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result"
    Write-Host "Waiting 30 seconds...`r`n"
    Start-Sleep -s 30
}
Catch {
    $_.Exception.ToString()
}

# Stop monitoring all database instances.
Try {
    Write-Host "Stopping all Monitors..."
    $command = @{"command" = "STOP"} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
    $dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result"
    Write-Host "Waiting 30 seconds...`r`n"
    Start-Sleep -s 30
}
Catch {
    $_.Exception.ToString()
}

# This will print out data like this:
Starting all Monitors...
Result: SUCCESS
Waiting 30 seconds...

Stopping all Monitors...
Result: SUCCESS
Waiting 30 seconds...
```

## Update the user password for a monitored database instance

```
$databaseId = 1
$monitorURL = $baseURL + "databases/$databaseId/update-password"
Try {
    Write-Host "Update the Monitor password for database $databaseId..."
    $command = @{"password" = "NewPassword!"} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
    $dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result`r`n"
}
Catch {
    $_.Exception.ToString()
}

# This will print out data like this:
Update the Monitor password for database 1...
Result: SUCCESS
```

## License Allocation examples

The examples below show how to use all License Allocation calls.

### Get information about currently installed licenses

```
$licenseURL = $baseURL + "databases/licenses/installed"

Try {
    Write-Host "Getting Installed license information with total amounts available for
    use and total amounts used..."
    $licenseListJSON = Invoke-RestMethod -Method Get -Uri $licenseURL -Headers $dpaHeader
    -TimeoutSec 60
    $licenseList = $licenseListJSON.data
    $licenseList | Format-Table -AutoSize
}
Catch {
    $_.Exception.ToString()
}

# This will print out data like this:
```

```
Getting Installed license information with total amounts available for use and total
amounts used...
```

licenseProduct	licenseCategory	licensesAvailable	licensesConsumed
DPACAT1	DPA_DB	100	22
DPACAT2	DPA_DB	100	16
DPAAzureSQL	DPA_DB	0	0
DPAVM	DPA_VM	100	12

## Get license information for a single database instance

```
$databaseId = 1
$licenseURL = $baseURL + "databases/$databaseId/licenses"
Try {
    Write-Host "Getting current license information for the database instance with
database ID of $databaseId."
    $licenseListJSON = Invoke-RestMethod -Method Get -Uri $licenseURL -Headers $dpaHeader
    -TimeoutSec 60
    $licenseInfo = $licenseListJSON.data
    $licenseInfo | Format-Table -AutoSize
}
Catch {
    $_.Exception.ToString()
}

# This will print out data like this:
Getting current license information for the database instance with database ID of 1.

serverName overLicensed vmLicenseProduct performanceLicenseProduct
-----
DEV-DPA                False DPAVM                DPACAT2
```

## Update license information for a database instance

```
$databaseId = 1
$licenseURL = $baseURL + "databases/$databaseId/licenses"

# Add a DPACAT2 and a DPAVM license
$licenseAllocation = @{"performanceLicenseProduct" = "DPACAT2";
```

```
                "vmLicenseProduct" = "DPAVM"} | ConvertTo-Json
Try {
    Write-Host "Updating license for database id $databaseId..."
    $licenseResultJSON = Invoke-RestMethod -Method Put -Uri $licenseURL -Body
$licenseAllocation -Headers $dpaHeader -TimeoutSec 60
    $licenseResult = $licenseResultJSON.data
    Write-Host "New License Allocation result for the database instance with database ID
of $databaseId."
    $licenseResult | Format-Table -AutoSize
}
Catch {
    $_.Exception.ToString()
}

# Remove the DPAVM license
$licenseAllocation = @{"performanceLicenseProduct" = "DPACAT2";
                "vmLicenseProduct" = "REMOVE"} | ConvertTo-Json
Try {
    Write-Host "Updating license for database id $databaseId..."
    $licenseResultJSON = Invoke-RestMethod -Method Put -Uri $licenseURL -Body
$licenseAllocation -Headers $dpaHeader -TimeoutSec 60
    $licenseResult = $licenseResultJSON.data
    Write-Host "New License Allocation result for the database instance with database ID
of $databaseId."
    $licenseResult | Format-Table -AutoSize
}
Catch {
    $_.Exception.ToString()
}

# This will print out data like this:
Updating license for database id 1...
New License Allocation result for the database instance with database ID of 1.

serverName overLicensed vmLicenseProduct performanceLicenseProduct
-----
DEV-DPA           False DPAVM           DPACAT2

Updating license for database id 1...
New License Allocation result for the database instance with database ID of 1.
```

```
serverName overLicensed vmLicenseProduct performanceLicenseProduct
-----
DEV-DPA                False                DPACAT2
```

## Annotation examples

The examples below show how to use all Annotation calls.

### Get a list of annotations for the last 30 days

```
$databaseId = 1
$annotationURL = $baseURL + "databases/$databaseId/annotations"

# Dates are in ISO 8601 format ( 2018-12-31T12:00:00.000-07:00 )
$endTime = Get-Date
$startTime = $endTime.AddDays(-30)
$startTime = [System.Web.HttpUtility]::UrlEncode($startTime.ToString("yyyy-MM-ddTHH:mm:ss.fffzzz"))
$endTime = [System.Web.HttpUtility]::UrlEncode($endTime.ToString("yyyy-MM-ddTHH:mm:ss.fffzzz"))

$request = [System.UriBuilder]$annotationURL
$request.Query = "startTime=$startTime&endTime=$endTime"
$annotationURL = $request.Uri

Try {
    Write-Host "Getting Annotations for the last 30 days..."
    $annotationListJSON = Invoke-RestMethod -Method Get -Uri $annotationURL -Headers
    $dpaHeader -TimeoutSec 60
    $annotationList = $annotationListJSON.data
    $annotationList | Format-Table -AutoSize
}
Catch {
    handleError $Error[0]
}

# This will print out data like this:
Getting Annotations for the last 30 days...

id title                description                createdBy time
type
```



```

-- -----
--
 98 Test Title API Test Event created by DPA API DPA API 2018-12-04T18:13:04-07:00
Custom
 99 Test Title API Test Event created by DPA API DPA API 2018-12-04T18:14:27-07:00
Custom
100 Test Title API Test Event created by DPA API DPA API 2018-12-04T18:16:46-07:00
Custom
etc.

```

## Create a new annotation

```

$databaseId = 1
$annotationURL = $baseURL + "databases/$databaseId/annotations"
$createTime = Get-Date
# Dates are in ISO 8601 format, no millis ( 2018-12-31T12:00:00-07:00 )
$createTime = $createTime.ToString("yyyy-MM-ddTHH:mm:sszzz")
$body = @{"title" = "API Test Title";
          "description" = "API Test Description";
          "createdBy" = "Test API User";
          "time" = "$createTime"} | ConvertTo-Json

Try {
  Write-Host "Creating Annotation..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $annotationURL -Body $body -Method POST -
Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $annotationId = $dpaResponse.id
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Creating Annotation...

 id title          description          createdBy          time          type
-- -----
148 API Test Title API Test Description Test API User 2019-01-03T15:20:36-07:00 API

```

## Delete an annotation

```
$databaseId = 1
$annotationId = 148
$annotationURL = $baseURL + "databases/$databaseId/annotations/$annotationId"
Try {
    Write-Host "Deleting Annotation with id of $annotationID..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $annotationURL -Method DELETE -Headers
    $dpaHeader -TimeoutSec 60
    Write-Host "Annotation with id of $annotationID deleted`r`n"
}
Catch {
    $_.Exception.ToString()
}

# This will print out data like this:
Deleting Annotation with id of 148...
Annotation with id of 148 deleted
```

## Database Registration examples

The examples below show how to use all Database Registration calls.

### Register and unregister a SQL Server database instance for monitoring

This example registers a new SQL Server database instance, waits 60 seconds, and then unregisters the database instance.

```
#-----
# Register a SQL Server database instance for monitoring.
#-----
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "SQLSERVER";
    "serverName" = "127.0.0.1";
    "port" = "1433";
    "sysAdminUser" = "sa";
    "sysAdminPassword" = "Password";
    "monitoringUser" = "dpa_test_m";
    "monitoringUserPassword" = "Password";
    "monitoringUserIsNew" = $true;
    "displayName" = "DPA_SQL2K12"} | ConvertTo-Json

Try {
```

```
Write-Host "Registering Database..."
$dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST -
Headers $dpaHeader -TimeoutSec 60
$dpaResponse = $dpaResponseJSON.data
$dpaResponse | Format-Table -AutoSize
$newDbId = $dpaResponse.databaseId
}
Catch {
    $_.Exception.ToString()
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#-----
# Un-register the SQL Server database instance.
#-----
if ($newDbId) {
    $registrationURL = $baseUrl + "databases/unregister-monitor"
    $body = @{"databaseId" = $newDbId;
              "removeMonitoringUser" = $true;
              "removeDatabaseObjects" = $true;
              "sysAdminUser" = "sa";
              "sysAdminPassword" = "Password"} | ConvertTo-Json
    Try {
        Write-Host "Registering Database..."
        $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST
-headers $dpaHeader -TimeoutSec 60
        $dpaResponse = $dpaResponseJSON.data
        $dpaResponse | Format-Table -AutoSize
    }
    Catch {
        $_.Exception.ToString()
    }
}

# This will print out data like this:
Registering Database...

databaseId result
-----
```

```

70 SUCCESS

Waiting 60 seconds...

Unregistering Database...

databaseId result
-----
70 SUCCESS

```

## Register and unregister an Oracle database instance for monitoring

This example registers a new Oracle database instance, waits 60 seconds, and then unregisters the database instance.

```

#-----
# Register an Oracle database instance for monitoring.
#-----
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "ORACLE";
          "serverName" = "127.0.0.1";
          "serviceNameOrSID" = "DPA_ORA11R1";
          "port" = "1521";
          "sysAdminUser" = "system";
          "sysAdminPassword" = "Password";
          "sysPassword" = "Password";
          "monitoringUser" = "dpa_test_m";
          "monitoringUserPassword" = "Password";
          "monitoringUserIsNew" = $true;
          "monitoringUserTableSpace" = "USERS";
          "monitoringUserTempTableSpace" = "TEMP";
          "oracleEBusinessEnabled" = $false;
          "displayName" = "DPA_ORA11R1"} | ConvertTo-Json

Try {
    Write-Host "Registering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST -
Headers $dpaHeader
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    $newDbId = $dpaResponse.databaseId
}

```

```
Catch {
    $_.Exception.ToString()
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#-----
# Un-register the Oracle database instance.
#-----
if ($newDbId) {
    $registrationURL = $baseUrl + "databases/unregister-monitor"
    $body = @{"databaseId" = $newDbId;
              "removeMonitoringUser" = $true;
              "removeDatabaseObjects" = $true;
              "sysAdminUser" = "system";
              "sysAdminPassword" = "Password"} | ConvertTo-Json
    Try {
        Write-Host "Unregistering Database..."
        $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST
    -Headers $dpaHeader -TimeoutSec 60
        $dpaResponse = $dpaResponseJSON.data
        $dpaResponse | Format-Table -AutoSize
    }
    Catch {
        $_.Exception.ToString()
    }
}

# This will print out data like this:
Registering Database...

databaseId result
-----
          71 SUCCESS

Waiting 60 seconds...

Unregistering Database...

databaseId result
```

```
-----
71 SUCCESS
```

## Register and unregister a MySQL database instance for monitoring

This example registers a new MySQL database instance, waits 60 seconds, and then unregisters the database instance.

```
#-----
# Register a MySQL database instance for monitoring.
#-----
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "MYSQL";
          "serverName" = "127.0.0.1";
          "port" = "3306";
          "sysAdminUser" = "root";
          "sysAdminPassword" = "Password";
          "monitoringUser" = "dpa_test_m";
          "monitoringUserPassword" = "Password";
          "monitoringUserIsNew" = $true;
          "displayName" = "DPA_MYSQL56"} | ConvertTo-Json
Try {
    Write-Host "Registering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST -
Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    $newDbId = $dpaResponse.databaseId
}
Catch {
    $_.Exception.ToString()
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#-----
# Un-register the MySQL database instance.
#-----
if ($newDbId) {
    $registrationURL = $baseURL + "databases/unregister-monitor"
```

```
$body = @{"databaseId" = $newDbId;
  "removeMonitoringUser" = $true;
  "removeDatabaseObjects" = $true;
  "sysAdminUser" = "root";
  "sysAdminPassword" = "Password"} | ConvertTo-Json

Try {
  Write-Host "Registering Database..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST
-headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
}
Catch {
  $_.Exception.ToString()
}
}

# This will print out data like this:
Registering Database...

databaseId result
-----
72 SUCCESS

Waiting 60 seconds...

Unregistering Database...

databaseId result
-----
72 SUCCESS
```

## Database Custom Properties examples

The examples below show how to use Database Custom Properties calls. Custom property values can be included in [custom email templates](#) for alert notifications.

### Create a custom property

This script creates a custom property and defines its name and description.

```

$propertyName = "Location"
$propertyDescription = "Location of the database server"
$createPropertyURL = $baseURL + "databases/properties"
$body = @{"name" = $propertyName; "description" = $propertyDescription;} | ConvertTo-
Json

Try {
    Write-Host "Creating custom property ..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $createPropertyURL -Body $body -Method POST
-headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    $propertyId = $dpaResponse.id
} Catch {
    $_.Exception.ToString()
}

#This will print out data like this:
id      name      description
--      ----      -
10434  Location  Location of the database server

```

## Create a custom property value

This script creates a value for the custom property created by the previous script.

```

property_id = 1
property_value = "New York"
$createValueURL = $baseURL + "databases/properties/" + $propertyId + "/values"
$body = $propertyValue | ConvertTo-Json

Try {
    Write-Host "Creating custom property value ..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $createValueURL -Body $body -Method POST -
Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
} Catch {
    $_.Exception.ToString()
}

```



```
#This will print out data like this:
id value
-- ----
1  "New York"
```

## Assign a property value to a monitored database instance

This script assigns a property value to a monitored database instance.

```
property_id = 1
property_value_id = 1
$assignPropertyValueURL = $baseUrl + "databases/" + $databaseId + "/properties/" +
$propertyId + "/values/" + $propertyValueId;

Try {
    Write-Host "Assigning custom property value ..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $assignPropertyValueURL -Method POST -
Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    Write-Host "Custom property value assigned to the DB with ID: $databaseId`r`n"
} Catch {
    $_.Exception.ToString()
}

#This will print out data like this:
Custom property value assigned to the DB with ID: 1
```

## Get all information about properties

This script returns information about all custom properties and their values.

```
$getPropertiesURL = $baseUrl + "databases/properties?require=assignment"

Try {
    Write-Host "Getting custom property information ..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $getPropertiesURL -Method GET -Headers
$dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
} Catch {
```

```

    $_.Exception.ToString()
}

#This will print out data like this:
Getting custom property information ...
id name          description          values
unassigned
-- ----          -
-----
1 Location Location of the database server {@{id=1; value="New York"; assignment=[1]}}
{4, 8}

```

## Delete a custom property

This script deletes a custom property.

```

property_id = 1
$deletePropertyURL = $baseUrl + "databases/properties/" + $propertyId

Try {
    Write-Host "Deleting custom property ..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $deletePropertyURL -Method DELETE -Headers
    $dpaHeader -TimeoutSec 60
    Write-Host "Custom property with id of $propertyID deleted`r`n"
} Catch {
    $_.Exception.ToString()
}

#This will print out data like this:
Custom property with id of 1 deleted

```

## Full working script

The following script combines all of the examples shown above into a script that can be run.

```

#-----
# Examples:
# - Get an access token
# - Database Monitor Examples
#   - Get Monitor Information for a single database
#   - Start or Stop monitoring a database instance given its database ID

```

```
# - Get Monitor Information for all databases
# - Start monitoring for all database instances
# - Stop monitoring for all database instances
# - ERROR: Get Monitor Information for a database that doesn't exist
# - ERROR: Start a database that doesn't exist
# - Licensing Examples
# - Get the currently installed license information
# - Get License Information for a single database
# - Update License Information for a single database
# - Annotation Examples
# - Gets a List of annotations for the last 30 days
# - Create a new annotation
# - Delete an annotation
# - Registration Examples
# - Register a MySQL database instance for monitoring
# - Un-register the MySQL database instance
#-----

#-----
# Configure the variables below for the DPA Host
#-----
$baseUrl = "https://localhost:8124/iwc/api/"
$refreshToken = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."
$databaseId = 1
#-----
# Nothing to configure below this line
#-----

#-----
# Function to parse the Response Data from DPA and print
# out the error information
#-----
Function handleError ($thisError) {
    Write-Host "-----"
    ForegroundColor Red
    Write-Host "Caught Exception at line:" $_.InvocationInfo.ScriptLineNumber -
ForegroundColor Red
    if ($_.Exception.Response) {
        $streamReader = [System.IO.StreamReader]::new
($_.Exception.Response.GetResponseStream())
        $errResp = $streamReader.ReadToEnd()
```

```

    $streamReader.Close()
}
if ($errResp) {
    # This will format the JSON
    $errResp = $errResp | ConvertFrom-Json | ConvertTo-Json -Depth 100
    Write-Host $thisError.Exception.Message -ForegroundColor Red
    Write-Host "Response:`r`n$errResp" -ForegroundColor Red
}
else {
    Write-Host $_.Exception.ToString() -ForegroundColor Red
}
Write-Host "-----" -
ForegroundColor Red
}

#-----
# Adding certificate exception to prevent API errors
# Uncomment this if you are getting trust errors and would
# like to run with self-signed certificates.
#-----
# add-type @"
# using System.Net;
# using System.Security.Cryptography.X509Certificates;
# public class TrustAllCertsPolicy : ICertificatePolicy {
#     public bool CheckValidationResult(
#         ServicePoint srvPoint, X509Certificate certificate,
#         WebRequest request, int certificateProblem) {
#         return true;
#     }
# }
# @"
# [System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy

#-----
# Get an access token
#-----
$authTokenURL = $baseURL + "security/oauth/token"
$body = @"grant_type" = "refresh_token"
        "refresh_token" = "$refreshToken"
Try {
    Write-Host "Getting Access Token..."

```

```
$dpaAuthResponse = Invoke-RestMethod -Uri $authTokenURL -Method POST -Body $body
$dpaAuthResponse | Format-List
}
Catch {
    handleError $Error[0]
    Write-Host 'Error getting authentication token, cannot continue' -ForegroundColor Red
    return
}

# If successful we will create our headers to be used for all API calls
$tokenType = $dpaAuthResponse.token_type
$accessToken = $dpaAuthResponse.access_token
$dpaHeader = @{}
$dpaHeader.Add("Accept", "application/json")
$dpaHeader.Add("Content-Type", "application/json;charset=UTF-8")
$dpaHeader.Add("Authorization", "$tokenType $accessToken")

#-----
# Database Monitor Examples
#-----

# Get Monitor Information for a single database
$monitorURL = $baseURL + "databases/$databaseId/monitor-information"
Try {
    Write-Host "Get Monitor Information for database with id of $databaseId..."
    $monitorJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers $dpaHeader -
TimeoutSec 60
    $monitor = $monitorJSON.data
    $monitor | Format-Table -AutoSize
}
Catch {
    handleError $Error[0]
}

# Start or Stop monitoring a database instance given its database ID.
# If it is already running stop it and then restart it
# If it is not running start it and then stop it
$monitorURL = $baseURL + "databases/$databaseId/monitor-status"
if ($monitor.monitorState -eq "Monitor Running") {
    $changeCommand = "STOP"
    $revertCommand = "START"
```

```
}
elseif ($monitor.monitorState -eq "Monitor Stopped") {
    $changeCommand = "START"
    $revertCommand = "STOP"
}
Try {
    Write-Host "$changeCommand Monitor for database $databaseId..."
    $command = @{"command" = $changeCommand} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
$dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result"
    Write-Host "Waiting 15 seconds...`r`n"
    Start-Sleep -s 15

    Write-Host "$revertCommand Monitor for database $databaseId..."
    $command = @{"command" = $revertCommand} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
$dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result"
    Write-Host "Waiting 15 seconds...`r`n"
    Start-Sleep -s 15
}
Catch {
    handleError $Error[0]
}

# Get Monitor Information for all databases
$monitorURL = $baseURL + "databases/monitor-information"
Try {
    Write-Host "Get Monitor Information for all databases..."
    $monitorListJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers $dpaHeader
-TimeoutSec 60
    $monitorList = $monitorListJSON.data
    $monitorList | Format-Table -AutoSize

    # Keep a list of running or started monitors to be used later
    $runningIds = @()
    foreach ($monitor in $monitorList) {
        if ($monitor.monitorState -eq "Monitor Running" -or
```

```
$monitor.monitorState -eq "Monitor Start No License" -or
$monitor.monitorState -like '*Start*')
{
    $runningIds += $monitor.dbId
}
}
Write-Host "Running Monitors: $runningIds`r`n"
}
Catch {
    handleError $Error[0]
}

# Start monitoring all database instances.
$monitorURL = $baseURL + "databases/monitor-status"
Try {
    Write-Host "Starting all Monitors..."
    $command = @{"command" = "START"} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
    $dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result"
    Write-Host "Waiting 30 seconds...`r`n"
    Start-Sleep -s 30
}
Catch {
    handleError $Error[0]
}

# Stop monitoring all database instances.
Try {
    Write-Host "Stopping all Monitors..."
    $command = @{"command" = "STOP"} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
    $dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result"
    Write-Host "Waiting 30 seconds...`r`n"
    Start-Sleep -s 30
}
Catch {
    handleError $Error[0]
}
```

```
}

# Try to put it back the way we found it by restarting the ones that were running
$command = @{"command" = "START"} | ConvertTo-Json
foreach ($dbId in $runningIds) {
    Try {
        $monitorURL = $baseURL + "databases/$dbId/monitor-status"
        Write-Host "Starting Monitor for database $dbId..."
        $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
        $result = $monitorJSON.data
        Write-Host "Result: $result`r`n"
    }
    Catch {
        handleError $Error[0]
    }
}

# Update the monitor database user password (Un-comment to use)
# $monitorURL = $baseURL + "databases/$databaseId/update-password"
# Try {
#     Write-Host "Update the Monitor password for database $databaseId..."
#     $command = @{"password" = "NewPassword!"} | ConvertTo-Json
#     $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
#     $result = $monitorJSON.data
#     Write-Host "Result: $result`r`n"
# }
# Catch {
#     handleError $Error[0]
# }

# Try to cause some errors...

# Get Monitor Information for a database that doesn't exist
$monitorURL = $baseURL + "databases/-1/monitor-information"
Try {
    Write-Host "Get Monitor Information for invalid database..."
    $monitorJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers $dpaHeader -
TimeoutSec 60
}
```



```
Catch {
    handleError $Error[0]
}

# Start a database that doesn't exist
$monitorURL = $baseURL + "databases/-1/monitor-status"
Try {
    Write-Host "START Monitor for invalid database..."
    $command = @{"command" = "START"} | ConvertTo-Json
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -Headers
    $dpaHeader -TimeoutSec 60
}
Catch {
    handleError $Error[0]
}

#-----
# Licensing Examples
#-----

# Get the currently installed license information
$licenseURL = $baseURL + "databases/licenses/installed"

Try {
    Write-Host "Getting Installed license information with total amounts available for
    use and total amounts used..."
    $licenseListJSON = Invoke-RestMethod -Method Get -Uri $licenseURL -Headers $dpaHeader
    -TimeoutSec 60
    $licenseList = $licenseListJSON.data
    $licenseList | Format-Table -AutoSize
}
Catch {
    handleError $Error[0]
}

# Get License Information for a single database
$licenseURL = $baseURL + "databases/$databaseId/licenses"
Try {
    Write-Host "Getting current license information for the database instance with
    database ID of $databaseId."
    $licenseListJSON = Invoke-RestMethod -Method Get -Uri $licenseURL -Headers $dpaHeader
```

```

-TimeoutSec 60
  $licenseInfo = $licenseListJSON.data
  $licenseInfo | Format-Table -AutoSize
}
Catch {
  handleError $Error[0]
}

# This will Update License Information for a single database setting the
# Performance License and the VM License to what it currently is.
# It should succeed but it should make no changes.
$dbProduct = $licenseInfo.performanceLicenseProduct
$vmProduct = $licenseInfo.vmLicenseProduct
$licenseAllocation = @{"performanceLicenseProduct" = $dbProduct;
                      "vmLicenseProduct" = $vmProduct} | ConvertTo-Json

Try {
  Write-Host "Updating license for database id $databaseId..."
  $licenseResultJSON = Invoke-RestMethod -Method Put -Uri $licenseURL -Body
  $licenseAllocation -Headers $dpaHeader -TimeoutSec 60
  $licenseResult = $licenseResultJSON.data
  Write-Host "New License Allocation result for the database instance with database ID
of $databaseId."
  $licenseResult | Format-Table -AutoSize
}
Catch {
  handleError $Error[0]
}

#-----
# Annotation Examples
#-----

# Gets a List of annotations for the last 30 days
$annotationURL = $baseURL + "databases/$databaseId/annotations"

# Dates are in ISO 8601 format ( 2018-12-31T12:00:00-07:00 )
$endTime = Get-Date
$startTime = $endTime.AddDays(-30)
$startTime = [System.Web.HttpUtility]::UrlEncode($startTime.ToString("yyyy-MM-
ddTHH:mm:ss.fffzzz"))
$endTime = [System.Web.HttpUtility]::UrlEncode($endTime.ToString("yyyy-MM-

```

```
ddTHH\:mm\:ss.fffzzz"))

$request = [System.UriBuilder]$annotationURL
$request.Query = "startTime=$startTime&endTime=$endTime"
$annotationURL = $request.Uri

Try {
    Write-Host "Getting Annotations for the last 30 days..."
    $annotationListJSON = Invoke-RestMethod -Method Get -Uri $annotationURL -Headers
    $dpaHeader -TimeoutSec 60
    $annotationList = $annotationListJSON.data
    $annotationList | Format-Table -AutoSize
}
Catch {
    handleError $Error[0]
}

# Create a new annotation
# Dates are in ISO 8601 format, no millis ( 2018-12-31T12:00:00-07:00 )
$annotationURL = $baseURL + "databases/$databaseId/annotations"
$createTime = Get-Date
$createTime = $createTime.ToString("yyyy-MM-ddTHH\:mm\:sszzz")
$body = @{"title" = "API Test Title";
    "description" = "API Test Description";
    "createdBy" = "Test API User";
    "time" = "$createTime"} | ConvertTo-Json

Try {
    Write-Host "Creating Annotation..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $annotationURL -Body $body -Method POST -
    Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    $annotationId = $dpaResponse.id
}
Catch {
    handleError $Error[0]
}

# Delete an annotation
if ($annotationId) {
    $annotationURL = $baseURL + "databases/$databaseId/annotations/$annotationId"
```

```

Try {
    Write-Host "Deleting Annotation with id of $annotationID..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $annotationURL -Method DELETE -Headers
$dpaHeader -TimeoutSec 60
    Write-Host "Annotation with id of $annotationID deleted`r`n"
}
Catch {
    handleError $Error[0]
}
}

#-----
# Registration Examples
#-----

#-----
# Register a SQL Server database instance for monitoring.
#-----
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "SQLSERVER";
    "serverName" = "127.0.0.1";
    "port" = "1433";
    "sysAdminUser" = "sa";
    "sysAdminPassword" = "Password";
    "monitoringUser" = "dpa_test_m";
    "monitoringUserPassword" = "Password";
    "monitoringUserIsNew" = $true;
    "displayName" = "DPA_SQL2K12"} | ConvertTo-Json
Try {
    Write-Host "Registering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST -
Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    $newDbId = $dpaResponse.databaseId
}
Catch {
    handleError $Error[0]
}

Write-Host "Waiting 60 seconds...`r`n"

```

```
Start-Sleep -s 60

#-----
# Un-register the SQL Server database instance.
#-----
if ($newDbId) {
    $registrationURL = $baseUrl + "databases/unregister-monitor"
    $body = @{"databaseId" = $newDbId;
        "removeMonitoringUser" = $true;
        "removeDatabaseObjects" = $true;
        "sysAdminUser" = "sa";
        "sysAdminPassword" = "Password"} | ConvertTo-Json

    Try {
        Write-Host "Unregistering Database..."
        $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST
-headers $dpaHeader -TimeoutSec 60
        $dpaResponse = $dpaResponseJSON.data
        $dpaResponse | Format-Table -AutoSize
    }
    Catch {
        handleError $Error[0]
    }
}

#-----
# Register an Oracle database instance for monitoring.
#-----
$registrationURL = $baseUrl + "databases/register-monitor"
$body = @{"databaseType" = "ORACLE";
    "serverName" = "127.0.0.1";
    "serviceNameOrSID" = "DPA_ORA11R1";
    "port" = "1521";
    "sysAdminUser" = "system";
    "sysAdminPassword" = "Password";
    "sysPassword" = "Password";
    "monitoringUser" = "dpa_test_m";
    "monitoringUserPassword" = "Password";
    "monitoringUserIsNew" = $true;
    "monitoringUserTableSpace" = "USERS";
    "monitoringUserTempTableSpace" = "TEMP";
    "oracleEBusinessEnabled" = $false;
```

```

        "displayName" = "DPA_ORA11R1"} | ConvertTo-Json
Try {
    Write-Host "Registering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST -
Headers $dpaHeader
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    $newDbId = $dpaResponse.databaseId
}
Catch {
    handleError $Error[0]
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#-----
# Un-register the Oracle database instance.
#-----
if ($newDbId) {
    $registrationURL = $baseURL + "databases/unregister-monitor"
    $body = @{"databaseId" = $newDbId;
        "removeMonitoringUser" = $true;
        "removeDatabaseObjects" = $true;
        "sysAdminUser" = "system";
        "sysAdminPassword" = "Password"} | ConvertTo-Json
    Try {
        Write-Host "Unregistering Database..."
        $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST
-headers $dpaHeader -TimeoutSec 60
        $dpaResponse = $dpaResponseJSON.data
        $dpaResponse | Format-Table -AutoSize
    }
    Catch {
        handleError $Error[0]
    }
}

#-----
# Register a MySQL database instance for monitoring.
#-----

```

```
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "MYSQL";
    "serverName" = "127.0.0.1";
    "port" = "3306";
    "sysAdminUser" = "root";
    "sysAdminPassword" = "Password";
    "monitoringUser" = "dpa_test_m";
    "monitoringUserPassword" = "Password";
    "monitoringUserIsNew" = $true;
    "displayName" = "DPA_MYSQL56"} | ConvertTo-Json

Try {
    Write-Host "Registering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST -
Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    $newDbId = $dpaResponse.databaseId
}
Catch {
    handleError $Error[0]
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#-----
# Un-register the MySQL database instance.
#-----

if ($newDbId) {
    $registrationURL = $baseURL + "databases/unregister-monitor"
    $body = @{"databaseId" = $newDbId;
        "removeMonitoringUser" = $true;
        "removeDatabaseObjects" = $true;
        "sysAdminUser" = "root";
        "sysAdminPassword" = "Password"} | ConvertTo-Json

    Try {
        Write-Host "Unregistering Database..."
        $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method POST
-headers $dpaHeader -TimeoutSec 60
        $dpaResponse = $dpaResponseJSON.data
        $dpaResponse | Format-Table -AutoSize
```

```
}
Catch {
    handleError $Error[0]
}
}

#-----
# Custom Property Examples
#-----
# Create custom property

$propertyName = "Location"
$propertyDescription = "Location of the database server"
$createPropertyURL = $baseURL + "databases/properties"
$body = @{"name" = $propertyName; "description" = $propertyDescription;} | ConvertTo-
Json

Try {
    Write-Host "Creating custom property ..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $createPropertyURL -Body $body -Method POST
-headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    $propertyId = $dpaResponse.id
}
Catch {
    handleError $Error[0]
}

#Create value of the custom property

$propertyValue = "New York"
$createValueURL = $baseURL + "databases/properties/" + $propertyId + "/values"
$body = $propertyValue | ConvertTo-Json

Try {
    Write-Host "Creating custom property value ..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $createValueURL -Body $body -Method POST -
Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
```



```
$propertyValueId = $dpaResponse.id
}
Catch {
    handleError $Error[0]
}

#Assign property value to DB

$assignPropertyValueURL = $baseURL + "databases/" + $databaseId + "/properties/" +
$propertyId + "/values/" + $propertyValueId;

Try {
    Write-Host "Assigning custom property value ..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $assignPropertyValueURL -Method POST -
Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
    Write-Host "Custom property value assigned to the DB with ID: $databaseId`r`n"
}
Catch {
    handleError $Error[0]
}

#Get all information about properties (including DB assignment)

$getPropertiesURL = $baseURL + "databases/properties?require=assignment"

Try {
    Write-Host "Getting custom property information ..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $getPropertiesURL -Method GET -Headers
$dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
}
Catch {
    handleError $Error[0]
}

#Delete custom property

$deletePropertyURL = $baseURL + "databases/properties/" + $propertyId
```

```
Try {
  Write-Host "Deleting custom property ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $deletePropertyURL -Method DELETE -Headers
$dpaHeader -TimeoutSec 60
  Write-Host "Custom property with id of $propertyID deleted`r`n"
}
Catch {
  handleError $Error[0]
}

# End of script
```

# View and manage trusted certificates

You can view and manage the following types of trusted certificates in DPA:


- **Certificates in the DPA trust store**

DPA can use certificates in the DPA trust store to connect to any database instance or LDAP server. These certificates are used only by DPA.

You can [view, import, and delete](#) certificates in the DPA trust store.

- **Certificates in the Java trust store**

DPA can use certificates in the Java trust store to connect to any database instance or LDAP server. These certificates are not managed through DPA. You can use DPA to [view the alias and expiration date](#) of these certificates.

 The `cacerts` file that contains these certificates is included in the JDK installed with DPA. The `cacerts` file is replaced each time DPA is upgraded, and any changes to this file are overwritten.

- **DB certificates**

A DB certificate is associated with a specific PostgreSQL or EDB Postgres database instance, and DPA uses it to connect to that instance. They are not shared with database instances that they haven't been assigned to, and they are not used by other services such as LDAP. DB certificates are used only by DPA.

You can use DPA to [view, import, and remove](#) DB certificates.

## Manage trusted certificates in the DPA trust store

DPA can use certificates in the DPA trust store to connect to any database instance or LDAP server. These certificates are used only by DPA. You can use DPA to [view, import, and delete](#) these certificates.

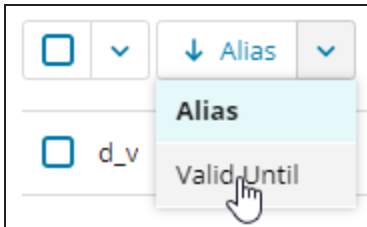
The DPA trust store is located, by default, in the `<DPA_install_dir>\iwc\tomcat\ignite_config\security` directory.

## View information about certificates in the DPA trust store

1. On the DPA menu, click Options.
2. Under Administration > Configuration, click Trusted Certificate Management.

The Trusted Certificate Management page opens. The DPA Trust Store tab lists the alias and expiration data of each certificate in the DPA trust store. You can:

- Sort by alias or expiration date. Click the down arrow to select the attribute you want to sort by. Click the sort button to reverse the sort order.



- Enter a string in the search field to search for a certificate by alias name. The list is filtered to show only certificates whose name includes the search string.



## Import a certificate into the DPA trust store

You can import a file that contains a single certificate, or you can import an entire keystore.

1. [Open](#) the DPA Trust Store tab.
2. Click Import certificate(s).

The Import certificate file dialog box opens.

## Import certificate file ✕

**Certificate file**

No file selected yet.

File types: PEM, CRT, CA-BUNDLE, DER, CER, PFX, P12, JKS, JCEKS.  
Maximum size: 1 MB


**Certificate alias (optional)**

Enter an alias to identify a PEM/DER certificate in the keystore. If blank, the subject is used.

**Keystore password (optional)**


The password is used for PKCS#12 key store to read the content.

3. Click Browse and select the certificate file.
4. If you are importing a PKCS#12 file, enter the keystore password.  
If you are importing a PEM or DER file, enter a unique alias to identify this certificate.

 If you do not enter an alias, the subject from the certificate is used. If the alias is not unique within the DPA trust store, DPA appends a number to make it unique.

### Import certificate file ✕

**Certificate file**

  ✕

File types: PEM, CRT, CA-BUNDLE, DER, CER, PFX, P12, JKS, JCEKS.  
Maximum size: 1 MB

**Certificate alias** (optional)

Enter an alias to identify a PEM/DER certificate in the keystore. If blank, the subject is used.

**Keystore password** (optional)

The password is used for PKCS#12 key store to read the content.

Cancel Import

5. Click Import.

DPA displays the import status. To display the status of each certificate, click Show complete results.

## Delete a certificate from the DPA trust store

You can remove certificates that are expired or no longer needed.

1. [Open](#) the DPA Trust Store tab.
2. Select one or more certificates to delete.



3. Click Delete, and then click Delete in the confirmation dialog box.

The certificate is removed from the DPA trust store.

## View trusted certificates in the Java trust store

Certificates in the Java trust store can be used by any Java application. DPA can use these certificates to connect to any database instance or LDAP server.

These certificates are located (by default) at `<DPA_install_dir>\iwc\jre\lib\security\cacerts`, and they are not managed through DPA. You can use DPA to view the names and expiration dates of these certificates.

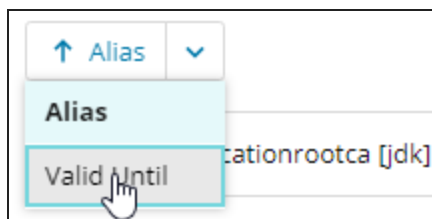
1. On the DPA menu, click Options.
2. Under Administration > Configuration, click Trusted Certificate Management.

The Trusted Certificate Management page opens.

3. Click Java Trust Store.

The Java Trust Store tab lists the alias and expiration date of each certificate in the Java trust store. You can:

- Sort by alias or expiration date. Click the down arrow to select the attribute you want to sort by. Click the sort button to reverse the sort order.



- Enter a string in the search field to search for a certificate by alias name. The list is filtered to show only certificates whose name includes the search string.



## Manage DB certificates

A DB certificate is associated with a specific PostgreSQL or EDB Postgres database instance, and DPA can use it to connect to that instance. DB certificates are used only by DPA, and they are stored in the DPA repository database.

You can use DPA to [view information](#) about DB certificates, [import](#) a DB certificate and associate it with a database instance, or [remove](#) a DB certificate.

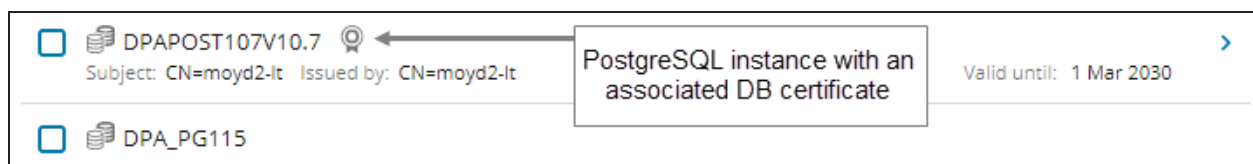
### View information about DB certificates

1. On the DPA menu, click Options.
2. Under Administration > Configuration, click Trusted Certificate Management.

The Trusted Certificate Management page opens.

3. Click DB Certificates.

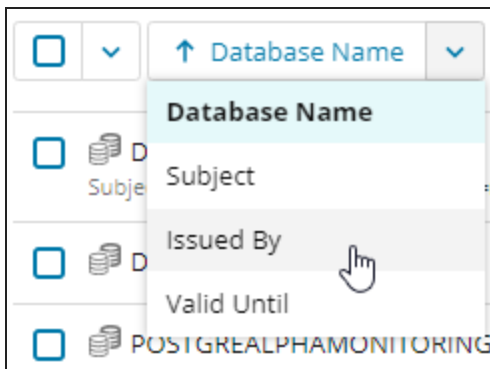
The DB Certificates tab lists all PostgreSQL or EDB Postgres database instances monitored by DPA. Instances that are associated with a DB certificate have a certificate icon next to the name, and information about the certificate is displayed.



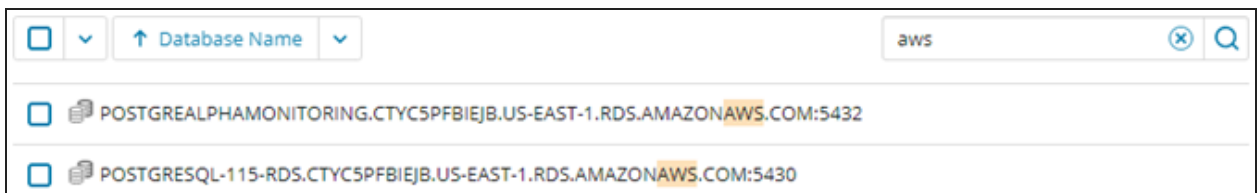


#### 4. You can:

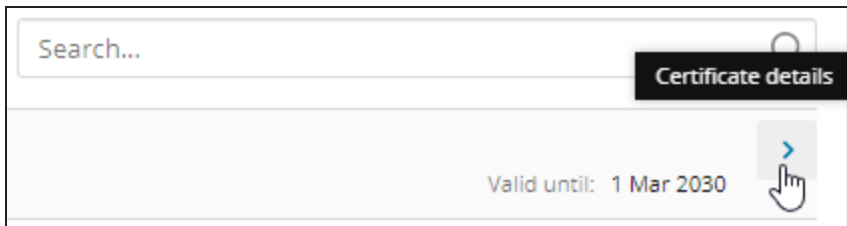
- Sort by database instance name, subject, issuer, or expiration date. Click the down arrow to select the attribute you want to sort by. Click the sort button to reverse the sort order.



- Enter a string in the search field to find database names, subjects, or certificates that include the string. The list is filtered to show only database instances that meet the search criteria.

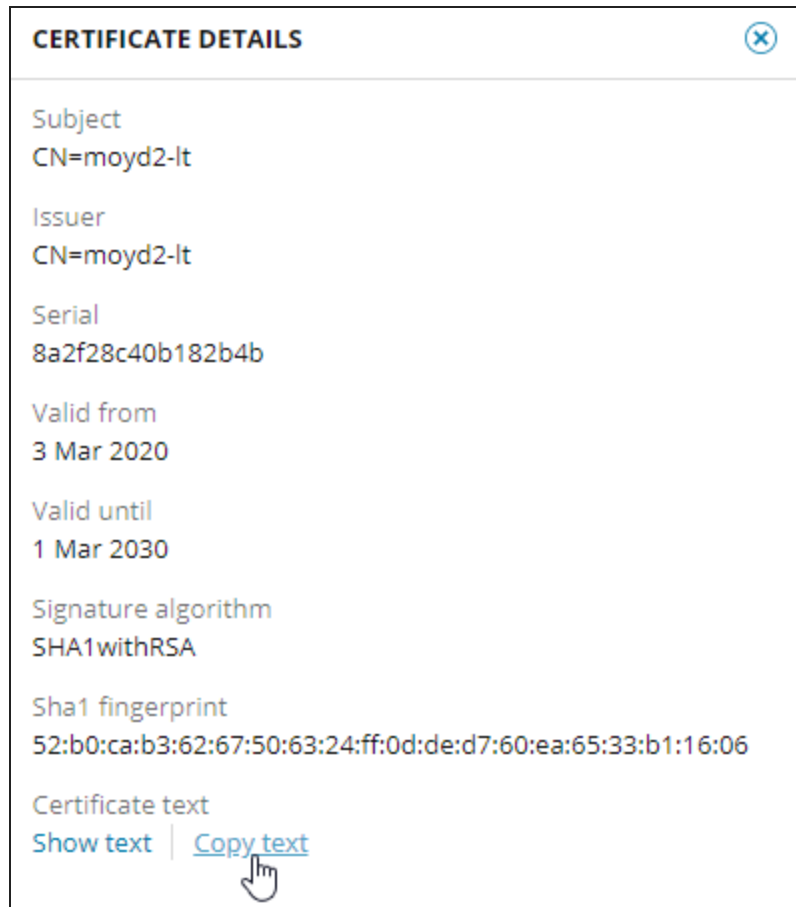


- Click the arrow on the right to view certificate details.



The Certificate Details panel opens. Click a link at the bottom to view or copy the

certificate text.

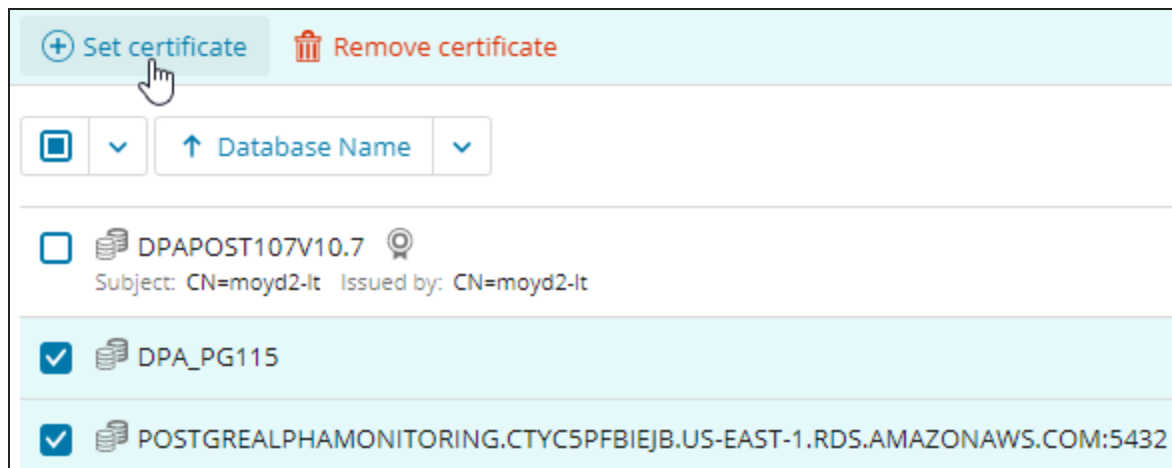


## Import a DB certificate and associate it with a PostgreSQL database instance

When you import a certificate through the DB Certificates tab, the certificate is stored in the DPA repository database. It is associated with one or more PostgreSQL database instances during the import process, and DPA uses it to connect to those instances.

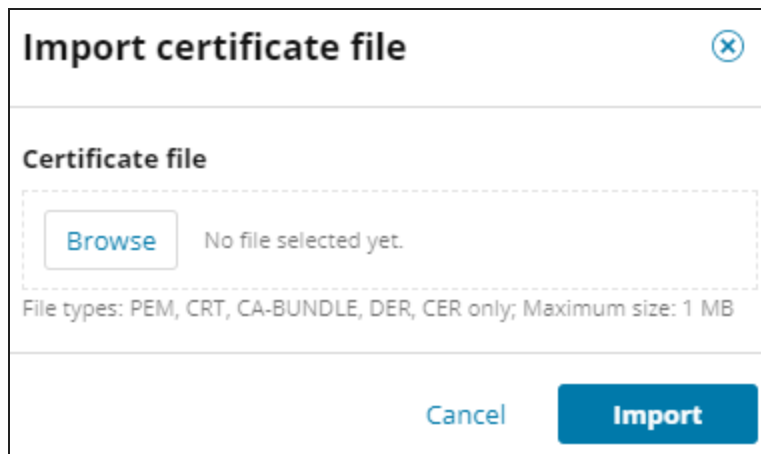
1. [Open](#) the DB Certificates tab.
2. Select the database instances you want to associate with a certificate.

The Set certificate and Remove certificate buttons are displayed.

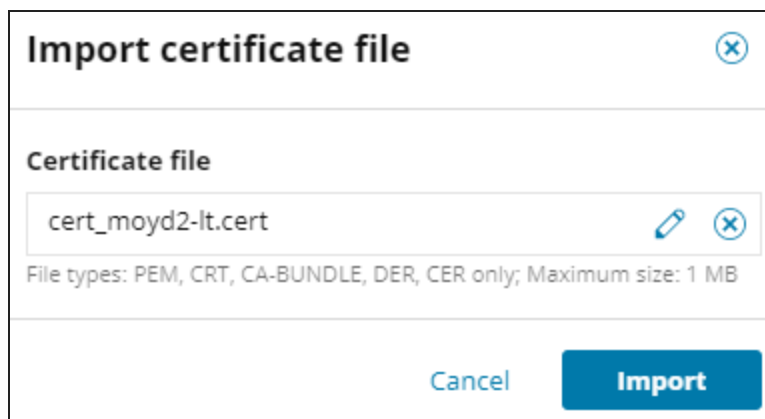


3. Click Set certificate.

The Import certificate file dialog box opens.



4. Click Browse and select the certificate file.



5. Click Import.

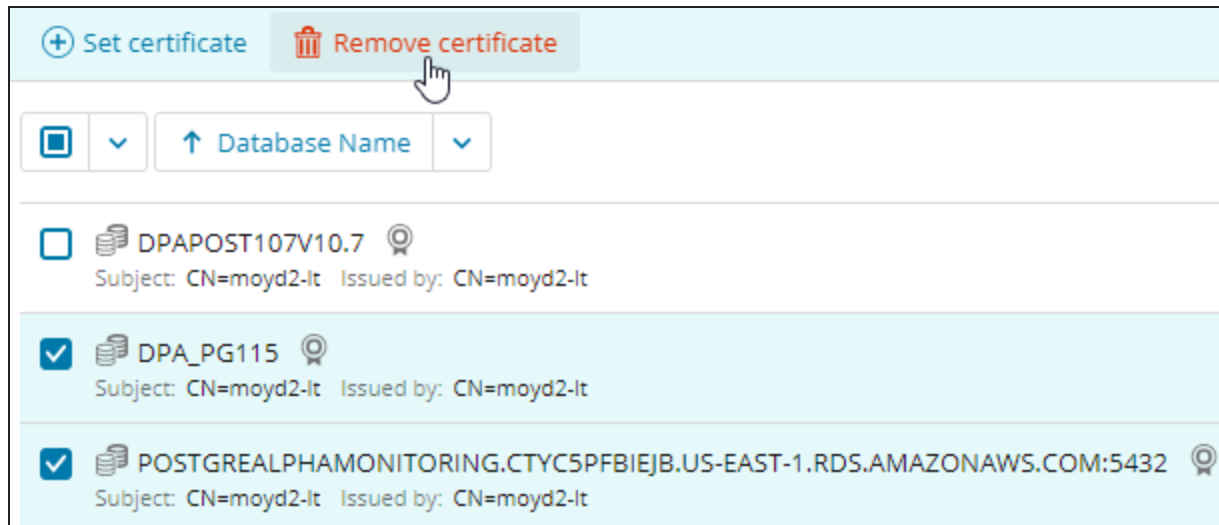
The certificate is imported and associated with the selected database instances.

## Remove a DB certificate

You can disassociate a DB certificate from a database instance and remove it from the DPA repository database. For example, you can remove a certificate when it expires.

1. [Open](#) the DB Certificates tab.
2. Select the database instances that are associated with certificates you want to remove.

The Set certificate and Remove certificate buttons are displayed.



3. Click Remove certificate, and then click Remove on the confirmation dialog box.

The selected database instances no longer have certificates associated with them, and the certificates are removed from the DPA repository database.

# DPA administrative tasks

See the following topics for information about administrative tasks in DPA:

- [Stop and start DPA](#)
- [Set advanced DPA options](#)
- [Enable SNMP monitoring in SCOM](#)
- [Configure password protection for DPA features that allow custom SQL](#)

## Stop and start DPA

### Stop and start DPA on a Windows server

- To stop DPA, run the following script file from the DPA directory:

```
shutdown.bat
```

For example, using the default DPA directory:

```
C:\Program Files\SolarWinds\DPA\shutdown.bat
```

- To restart DPA, run the following script file from the DPA directory:

```
startup.bat
```

For example, using the default DPA directory:

```
C:\Program Files\SolarWinds\DPA\startup.bat
```

Alternatively, use the Windows Control Panel to stop the Ignite PI Server service. To restart DPA, start the service again.

### Stop and start DPA on a Linux server

- To stop DPA, run the following command from the DPA directory:

```
shutdown.sh
```

For example, using the default DPA directory:

```
/home/solarwinds/dpa_v_v/shutdown.sh
```

- To restart DPA, run the following command from the DPA directory:

```
startup.sh
```

For example, using the default DPA directory:

```
/home/solarwinds/dpa_v_v/startup.sh
```

## Set advanced DPA configuration options

You can use advanced options to change DPA's default behavior. For example, you can:

- Change the [default expiration times](#) for access and refresh tokens.
- Change the [Warning and Critical thresholds](#) for anomaly detection.
- Change the days included in [baseline calculations](#).

In most cases, you will change these options when instructed to do so by SolarWinds Support or based on information from another topic of this administration guide (such as the linked topics above).

1. On the DPA menu, click Options.
2. Under Administration > Configuration, click Advanced Options.


The System Options tab lists options that apply to all database instances. The list includes a description of each option.

3. If you are setting an option that applies to a single database instance, click DB Instance Options and select the database instance.
4. Click the name of an option to open the Edit Option dialog.
5. To change an option value, enter the New Value and click Update.

## Enable SNMP Monitoring in SCOM

You can set up DPA to use SNMP to monitor System Center Operations Manager (SCOM).

1. On the DPA menu, click Options.
2. Under Administration > Users & Contacts, click Contact Management.
3. Click Create SNMP Contact.
4. Enter the SCOM host IP address and port in the Trap Receiver fields. The default port is 162.
5. Enter the community string that was set up on the SNMP Service on the SCOM host.

 This string is case sensitive.

6. On the DPA server, make sure the SNMP service is running and the community string set matches the string you entered in the SNMP Contact window.

# Configure password protection for DPA features that allow custom SQL

To prevent unauthorized users from entering malicious SQL, you can configure password protection for DPA features that allow users to enter custom SQL. These features include:

- Custom metrics
- Custom alerts
- The database query tool
- The Update DB Instance Connection Wizard

## Enable password protection

When password protection is enabled, users are prompted for the specified password when they test or save a custom metric or custom alert, and when they open the database query tool or the Updated DB Instance Connection Wizard.

1. Open the `sqlauth.xml` file in a text editor. This file is located in the following directory:

```
<DPA-install-dir>\iwc\tomcat\ignite_config\iwc\security
```

The default location is:

```
C:\Program Files\SolarWinds\DPA\iwc\tomcat\ignite_config\iwc\security
```

2. Enter the password as the value of the `<entry key="sql.authentication.password">` setting. The password must contain:
  - At least 7 characters
  - At least 1 numeric character

For example:

```
<entry key="sql.authentication.password">MyPassword1</entry>
```

If the password includes special characters (for example, `&`), enclose the password with CDATA as follows:

```
<entry key="sql.authentication.password"><![CDATA[My&Password1]]></entry>
```

3. Save the file.

Changes take effect immediately. The password in the `sqlauth.xml` file is encrypted the first time DPA prompts a user to enter it.

## Disable password protection

Password protection is disabled by default. If you enable it and then wish to disable it again, complete the following steps.

1. Open the `sqlauth.xml` file in a text editor. This file is located in the following directory:

```
<DPA-install-dir>\iwc\tomcat\ignite_config\iwc\security
```

The default location is:

```
C:\Program Files\SolarWinds\DPA\iwc\tomcat\ignite_config\iwc\security
```

2. Remove the value of the `<entry key="sql.authentication.password">` setting. For example:

```
<entry key="sql.authentication.password"></entry>
```

3. Save the file.

Changes take effect immediately.