Documentation

AppDynamics for Databases



1. Install and Get Started with AppDynamics for Databases 2
  1.1 Installation Requirements 2
    1.1.1 Required Database Permissions 4
  1.2 Install AppDynamics for Databases 9
    1.2.1 Enabling AppDynamics for Databases Security 10
  1.3 Start AppDynamics for Databases 15
  1.4 Add a License 16
  1.5 Add a Collector 17
    1.5.1 Infrastructure Monitoring (Monitor OS?) 22
  1.6 Use the Database Platform Window 24
  1.7 Verify Collector Setup 34
  1.8 Integrate AppD4DB with AppDynamics Pro 34
    1.8.1 Integrate AppD4DB with AppDynamics Pro 3.7 35
    1.8.2 Integrate AppD4DB with AppDynamics Pro 3.6 36
    1.8.3 Installing AppD4DB and AppD Controller on the Same Host 37
  1.9 Use AppDynamics Pro with AppDynamics for Databases 38

Copyright ©AppDynamics 2013

Install and Get Started with AppDynamics for Databases

AppDynamics for Database (AppD4DB) can monitor three categories of collectors. There are several collector types within these categories. The following lists the supported databases, servers, and storage systems now supported:

- **Database Collector**: DB2, MySQL, Oracle, PostgreSQL, Sybase ASE, and Sybase IQ
- **Server Collector**: Linux Server, Solaris Server, and Windows Server
- **Storage Collector**: NetApp

The following topics help you get AppD4DB up and running and monitoring:

- Installation Requirements
  - Required Database Permissions
  - Infrastructure Monitoring (Monitor OS?)
- Install AppDynamics for Databases
- Start AppDynamics for Databases
- Add a License
- Add a Collector
- Verify Collector Setup
- Integrate AppD4DB with AppDynamics Pro
  - Integrate AppD4DB with AppDynamics Pro 3.7
  - Integrate AppD4DB with AppDynamics Pro 3.6
- Use AppDynamics Pro with AppDynamics for Databases

Installation Requirements

For large scale deployments, or where monitored systems are extremely busy, please contact AppDynamics Support for installation recommendations and assistance.

- Systems that AppDynamics for Database Monitors
- Hardware Requirements
- Network Requirements
- Operating System Compatibilities
- Browser Compatibilities
  - GUI Performance Considerations

Systems that AppDynamics for Database Monitors

Once AppD4DB is up and running, you can create collectors that run on the AppD4DB server to monitor any of the following systems:

- **Databases**: DB2, MySQL, Oracle, PostgreSQL, Sybase ASE, and Sybase IQ
- **Servers**: Linux Server, Solaris Server, or Windows Server
- **Storage**: NetApp

AppD4DB collectors are not installed on the monitored systems.

Hardware Requirements

AppDynamics for Databases can be installed on physical or virtual machines.

You should install AppDynamics for Databases on a dedicated machine.

- Installation package: 142 MB Windows, 140 MB Linux
- Disk space usage after installation: 614 MB
- Disk space required for historical data storage:
  - 100 MB per database collector (for data retention period = 1 week)
  - 10 MB per server collector (for data retention period = 1 week)
- Minimum recommended hardware:
  - 1-10 collectors: 2 GB RAM, Single CPU, 1 disk
  - 10-20 collectors: 4 GB RAM, 2 CPUs, 1 disk
  - More than 20 collectors: 8 GB RAM, 4 CPUs, fast disk
- Minimum Memory requirements:
  - 512 MB for AppDynamics for Databases Repository
  - 512 MB AppDynamics for Databases GUI
  - 100 MB per collector

For information on hardware requirements of the AppDynamics Controller, see AppDynamics Pro, Controller System Requirements.

Network Requirements

- The machine on which the database is running or the machine you want to monitor must be accessible from the AppDynamics for Databases installation machine. This machine must have a network connection, internet or intranet.
- If there is a firewall, the database listener port (and optionally the SSH or WMI port) must be open.

Operating System Compatibilities

- Microsoft Windows platforms, including Windows 7 and Windows 8
- All Linux versions/platforms

Browser Compatibilities

For better navigation and performance, use Chrome or Safari.

- Google Chrome 1.x and later
- Safari 4.x and later

- Mozilla FireFox 3.x and later
- Internet Explorer 7.x and later

GUI Performance Considerations

The charts and reports within AppDynamics for Databases are created with JavaScript; consequently, the speed at which they are rendered depends on the performance of the JavaScript engine within the Browser. For this reason, although AppDynamics for Databases is supported on any Browser, we recommend using better performing browsers, such as Chrome or Safari.

Required Database Permissions

Each monitored database requires permissions for the AppDynamics for Databases (AppD4DB) user so that it can gather important monitoring data. The database user is specified when you are adding a collector. Before adding the collector, ensure a user for the collectoris available with the required permissions as stated below. The monitor user must be able to connect to the database remotely from the AppD4DB machine. The permissions required are database dependent.

- MySQL Database Permissions
- SQL Server Database Permissions
    - Minimum Permissions Required for SQL Server Authentication
- Oracle Database Permissions
- Sybase ASE Database Permissions
- Postgres SQL Database Permissions
- IBM DB2 (LUW) Database Permissions

MySQL Database Permissions

The MySQL user AppD4DB uses to monitor the MySQL database, must have "SELECT", "PROCESS", and "SHOW DATABASES" privileges on all databases.

If you do not have a suitable existing user, you can use a command such as the following to create a new user; where “host” is the hostname or IP address of the machine running AppD4DB, and “password” is a suitably secure password:

```
GRANT SELECT,PROCESS,SHOW DATABASES on *.* to 'appd4db'@'host' identified by
'password';
FLUSH privileges;
```

SQL Server Database Permissions

You can need a SQL Server Authenticated user that is a member of the sysadmin server role or a Windows Authenticated Account with sysadmin permissions. If you don't have access to a SQL Server Authenticated user that is a member of the sysadmin server role, you can use the procedure below to create a user with the minimum permissions required.

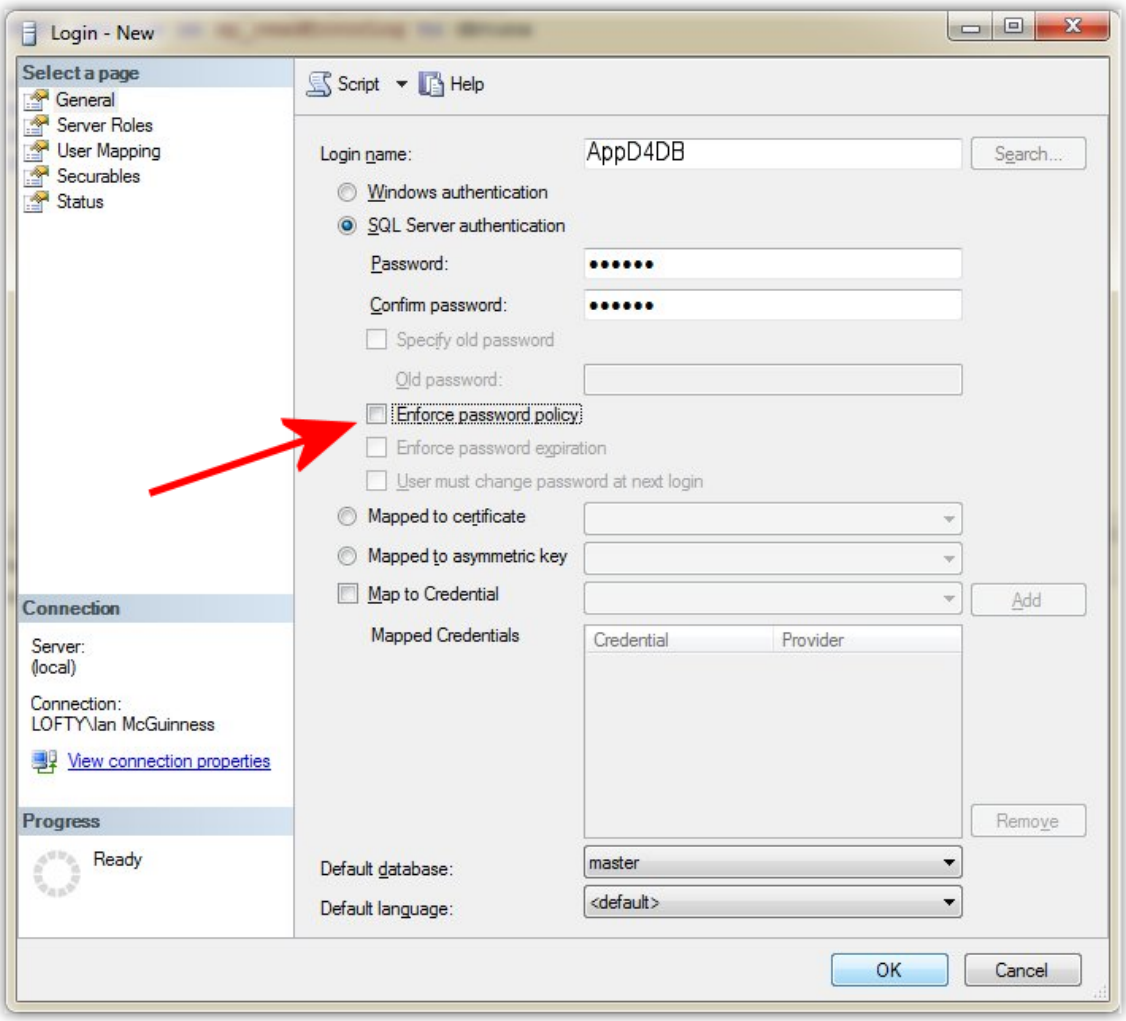
If you are using Windows Authentication, leave the username/password blank when you add the

collector. You then have to change the Logon account for the Window AppD4DB collector service. See Using Windows Authentication to Monitor SQL Server.

Minimum Permissions Required for SQL Server Authentication

Use the following to create a logon user that provides the minimal level of permissions required in order to gain full AppD4DB/SQL Server functionality.

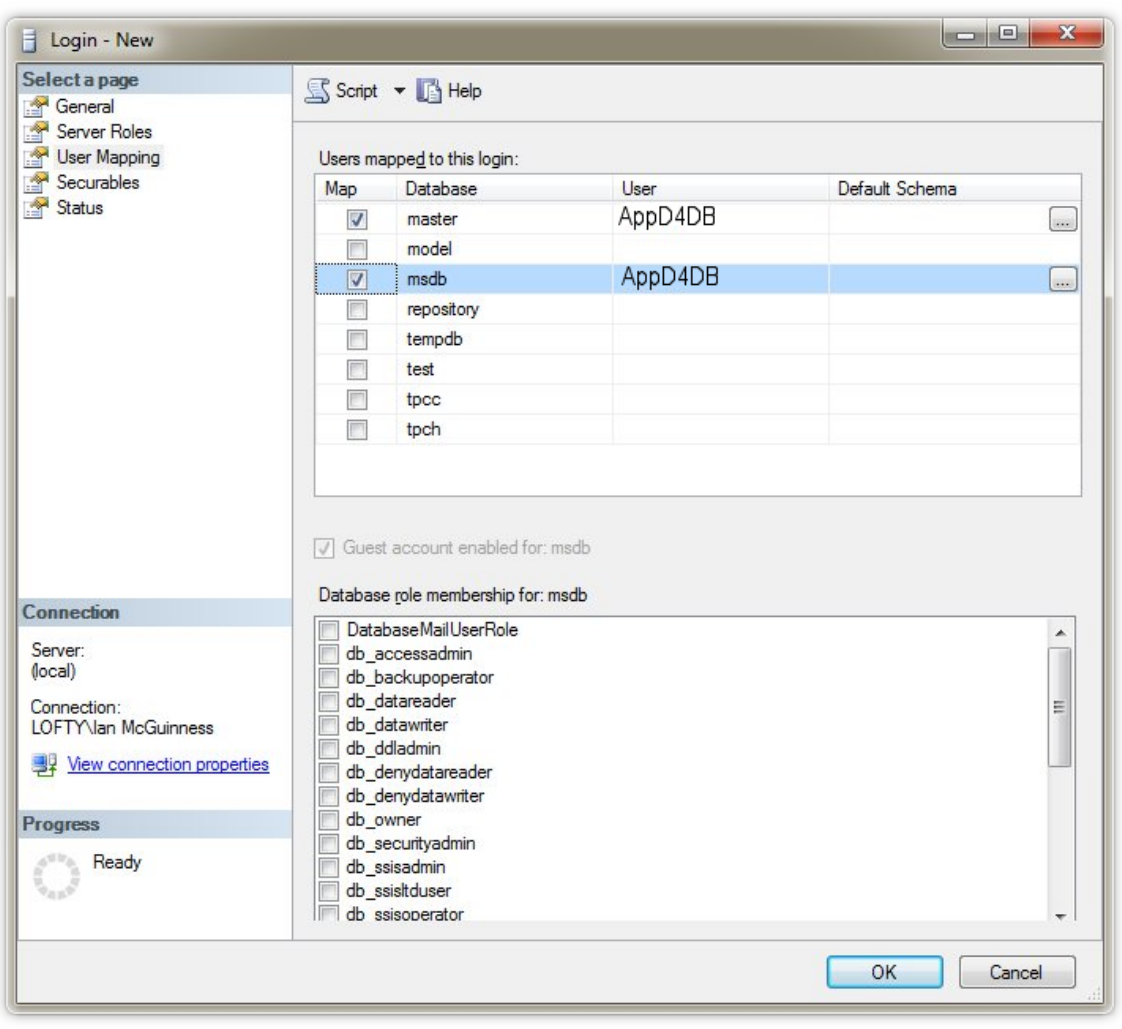
1. Using SQL Server Management Studio, create a new login for AppD4DB.



The user can be a Windows authenticated account (if monitoring via AppD4DB running on Windows) or SQL Server authenticated. The above example displays a SQL Server authenticated account.

Note: You must disable the **Enforced password policy** option as shown above. This account will login in a non-interactive manner (via the AppD4DB collector) and therefore will not be able to modify its password at login.

2. From the **User Mapping** tab, map the new user to the **master** and **msdb** databases.



3. Once you have created the login, give the following privileges to the user, substituting <userName> with the name you specified on the **Login - New window**:

Note: You can execute the following as a batch from a query window in Management Studio.

```
use master
GRANT VIEW ANY DATABASE TO <userName>
GRANT VIEW ANY definition TO <userName>
GRANT VIEW server state TO <userName>
GRANT SELECT ON sys.sysaltfiles TO <userName>
GRANT execute on sp_helplogins TO <userName>
GRANT execute on sp_readErrorLog TO <userName>
```

```
use msdb
GRANT SELECT on dbo.sysjobsteps  TO <userName>
GRANT SELECT on dbo.sysjobs  TO <userName>
GRANT SELECT on dbo.sysjobhistory  TO <userName>
```

Oracle Database Permissions

The permissions required for the Oracle user are:

- connect
- select any dictionary
- resource

To create a user with these permissions, you can run the following SQL. In this SQL, change "password" to a safe and secure password, and change the tablespace names, "users" and "temp" to those available in your Oracle instance.

```
create user appd4db identified by password
default tablespace users
temporary tablespace temp;

grant connect, select any dictionary, resource to appd4db;
```

Explain Plans:
AppD4DB can generate explain plans within its SQL drilldown window. To enable this functionality, you must have a plan table accessible to the AppD4DB schema user. You can create this plan table with the following command from sqlplus when logged on as the AppD4DB monitoring user:

Windows:

```
@?\rdbms\admin\utlxplan.sql
```

Linux:

```
@?/rdbms/admin/utlxplan.sql
```

Sybase ASE Database Permissions

For complete AppD4DB functionality, the monitoring user requires the sa_role and mon_role privilege.

To create a new dedicated user for AppDynamics for Database, you can use following sample

user creation script. Before running the script, change "password" to a more secure value.

```
exec  sp_addlogin  'appd4db', 'password', @defdb='master',
@deflanguage='us_english', @fullname='appd4db monitoring account', @auth_mech =
'ANY'
go
exec  sp_locklogin  'appd4db', 'unlock'
go
exec  sp_role 'grant', 'sa_role', 'appd4db'
go
exec  sp_role 'grant', 'mon_role', 'appd4db'
go
```

Also, the following configuration parameters must be set to 1 (true) in order to monitor the Sybase ASE database with AppD4DB: "enable monitoring", "wait event timing", "SQL batch capture", and "object lockwait timing". You should also set "max SQL text monitored" to at least 8192 (8kB).

Here is an example of the commands required to configure these settings:

```
sp_configure "enable monitoring", 1
go
sp_configure "wait event timing", 1
go
sp_configure "SQL batch capture", 1
go
sp_configure "object lockwait timing", 1
go
sp_configure "max SQL text monitored", 8192
go
```

If the value for "max SQL text monitored" was previously less than 4096, then increasing this setting will require that you restart the Sybase ASE instance.

Postgres SQL Database Permissions

The monitoring user must be superuser and must be able to connect remotely to the PostgreSQL instance from the AppD4DB machine. For information on how to allow users to authenticate from a remote client machine, see http://www.postgresql.org/docs/8.3/static/auth-pg-hba-conf.html.

For complete AppD4DB functionality, the "track_activities" parameter must be enabled on the PostgreSQL server.

IBM DB2 (LUW) Database Permissions

For complete AppD4DB functionality the following monitoring switches of the DB2 server need to be enabled: "BUFFERPOOL", "LOCK", "SORT", "STATEMENT", "TABLE", "TIMESTAMP", and "UOW".

You can enable these monitoring switches by the following commands:

```
update dbm cfg using dft_mon_uow on;
update dbm cfg using dft_mon_stmt on;
update dbm cfg using dft_mon_timestamp on;
update dbm cfg using dft_mon_lock on;
update dbm cfg using dft_mon_bufpool on;
update dbm cfg using dft_mon_table on;
update dbm cfg using dft_mon_sort on;
```

The monitoring user needs SYSMON authority and connect privileges to monitor. In general, this user must be a part of the sysmon_group.

Install AppDynamics for Databases

Prerequisite: Before installing, ensure your environment meets the requirements stated in Installation Requirements.

The installation process is simple and should take no longer than five minutes. You will be prompted to enter your license key when the install completes and the UI service starts. See also Password Protect the AppDynamics for Databases UI.

Install on Windows

Unzip and double-click the executable package, AppD-Database-x.x.x.x-Setup.exe to start the setup process.

The Windows processes for AppDynamics for Databases (AppD4DB) start at the end of the installation.

Install on Linux

Note: Do not install AppDynamics for Databases as the root user.

1. Use the following command to decompress the executable package:

```
gunzip <download_file>.tar.gz
```

2. Run the ./start.sh script from the installation directory to start the processes for AppD4DB.

Note: You should edit /etc/init.d to include running the AppD4DB startup script so the AppD4DB processes will start automatically on reboot.

Install on Linux in Solaris Environment

Note: Do not install AppDynamics for Databases as the root user.

Use the following command to decompress the executable package:

```
tar xvf  <download_file>.tar
```

Enabling AppDynamics for Databases Security

If you run AppDynamics for Databases on a publicly accessible server, or if you'd like to lock down its usage internally, then the simplest solution is to username/password protect access to the UI. You have the option to setup basic security, best for an environment where very few users will have access to the AppD4DB GUI, or you can integrate AppD4DB with your LDAP server to grant many users and groups access.

- Setup Basic Security
  - To implement authentication:
- Setup LDAP/Active Directory Integration
  - To complete the fields in the LDAP/Active Directory Authentication section
- Enable Authentication Tracking
- Monitor Access Attempts

Setup Basic Security

AppD4DB has three predefined users:

- admin: The administrator is has the permissions of all three security role resources: Administration Access, Read-Only Access, and masterUser Access. The **default password** for the admin user is "admin".
- readonly: The readonly user has the permissions of only the Read-Only Access Role. The **default password** for the readonly user is "readonly".
- master: The master user has the permissions of the appd4db-admin role. The **default password** for the master user is "welcome".

AppD4DB has three predefined security roles resources:

- Administration Access: Provides total control over AppDynamics for Databases. Users created with this role can perform all functions. The **default role name** for the Administration Access security role resource is "admin".
- Read-Only Access: Provides read-only access to AppD4DB. Users created with this role cannot add collectors, licenses or create alerts. This user has access to all the database, server, and NetApp activity pages and reports. The **default role name** for the Read-Only Access Access security role resource is "readonly".
- masterUser Access: Provides access only to the Security Setup page, where the master user can modify passwords for basic authentication, modify security role names, and can change the LDAP/Active Directory configuration. The **default role name** or the Read-Only Access Access security role resource is "readonly".

These users and roles are initially configured in <*AppDInstallDir* >\apache-tomcat\conf\tomcat-users.xml and <*AppDInstallDir*>\apache-tomcat\conf\web.xml, respectively. Passwords are encrypted. You can change the passwords and rolenames on the

AppD4DB Security window.

tomcat-users.xml

```
<tomcat-users>
<role rolename="admin"/>
<role rolename="readonly"/>
<role rolename="appd4db-admin"/>
<user
password="8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918"
roles="admin,readonly,appd4db-admin" username="admin"/>
<user
password="8171bacf32668a8f44b90087ad107ed63170f57154763ba7e44047bf9e5a7be3"
roles="readonly" username="readonly"/>
<user
password="280d44ab1e9f79b5cce2dd4f58f5fe91f0fbacdac9f7447dffc318ceb79f2d02"
roles="appd4db-admin" username="master"/>
</tomcat-users>
```

Note: Do not change the contents of tomcat-users.xml.

To implement authentication:

1. At the bottom of <*AppDInstallDir*>\apache-tomcat\conf\web.xml, look for the following code:



Note: Do not change the contents of web.xml except as instructed below.

2. Insert a closing XML comment tag after "<--Password protect AppDynamics for Database pages". The closing XML comment tag is "-->".

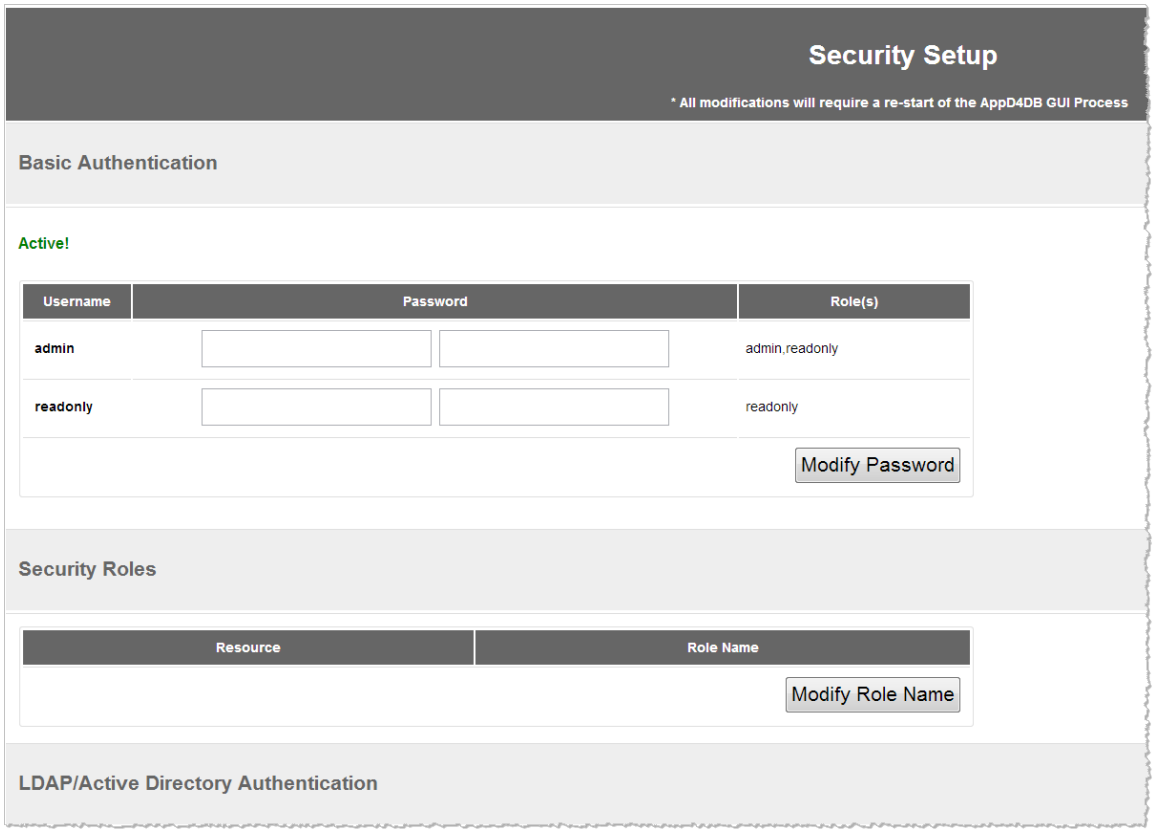
3. Remove the closing XML comment tag before "</web-app>". The closing XML comment tag is "-->".

4. Restart the AppD4DB UI service.

- Windows: Go to the Windows services manager and restart "DBTuna GUI".
- Linux: From the AppD4DB install dir, run the stop.sh script followed by the start.sh script.

5. In a browser, go to the security page. For example, http://<*hostname*>:8090/security.

The following dialog appears where you can setup basic security or enable LDAP/Active Directory Service integration for AppD4DB:

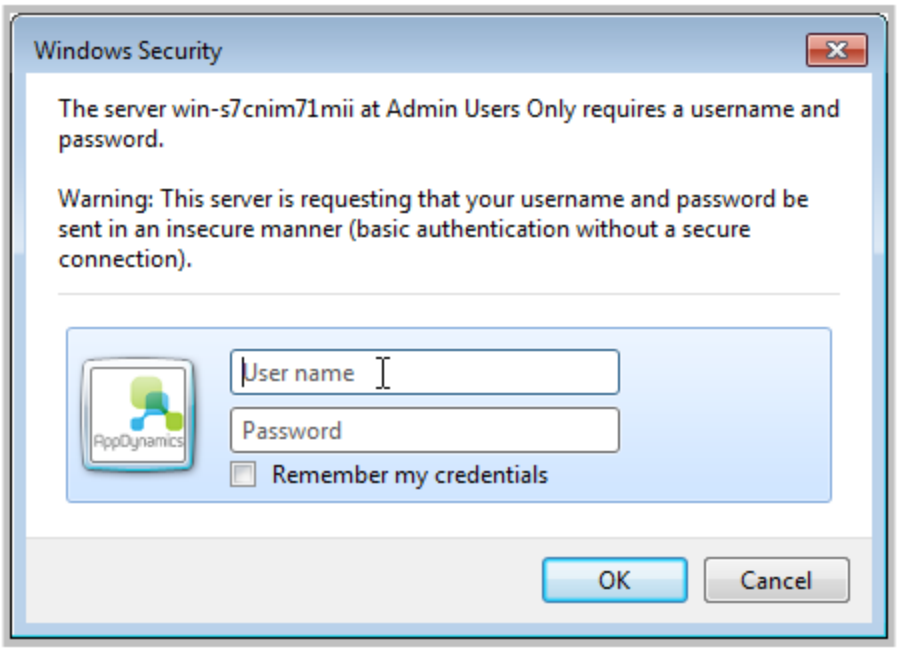


2. Enter the passwords for the admin and readonly users and then click **Modify Password**.

***To change the password of a user, enter the password twice in the boxes provided and then click *Modify Password**.

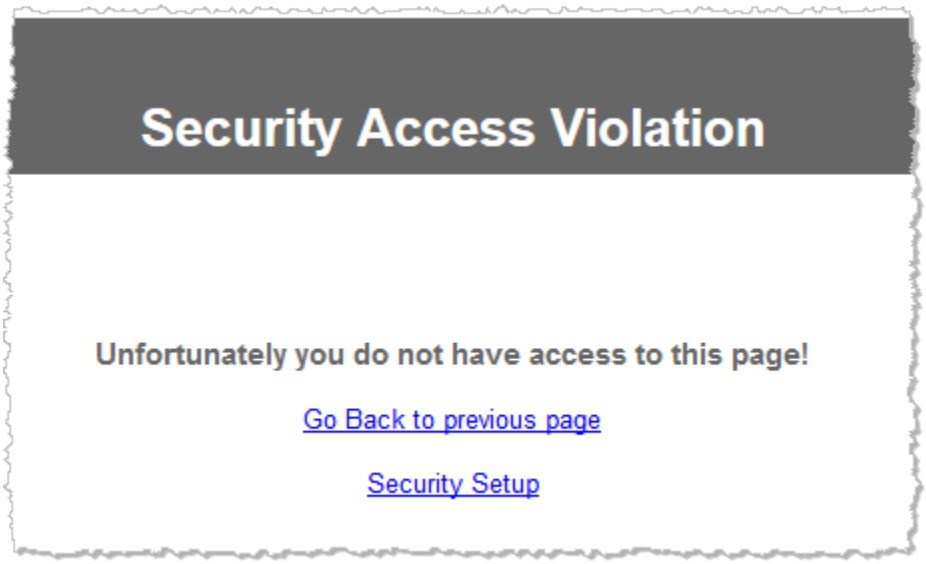
3. You can change the role name of any of the security roles resources by entering the new Role Name and then clicking **Modify Role Name**.

When you have security enabled, users must enter the security credentials to access the AppDynamics for Database (AppD4DB) GUI. The appearance of the logon dialog differs depending on the browser used to access the AppD4DB GUI. The following is the logon dialog as it appears in Windows Internet Explorer9.



If you enter the wrong username/password combination, the uncompleted logon dialog reappears so you can re-enter your credentials.

If you try to access a page not accessible to the role to which your username has been assigned, you will receive a security violation error.

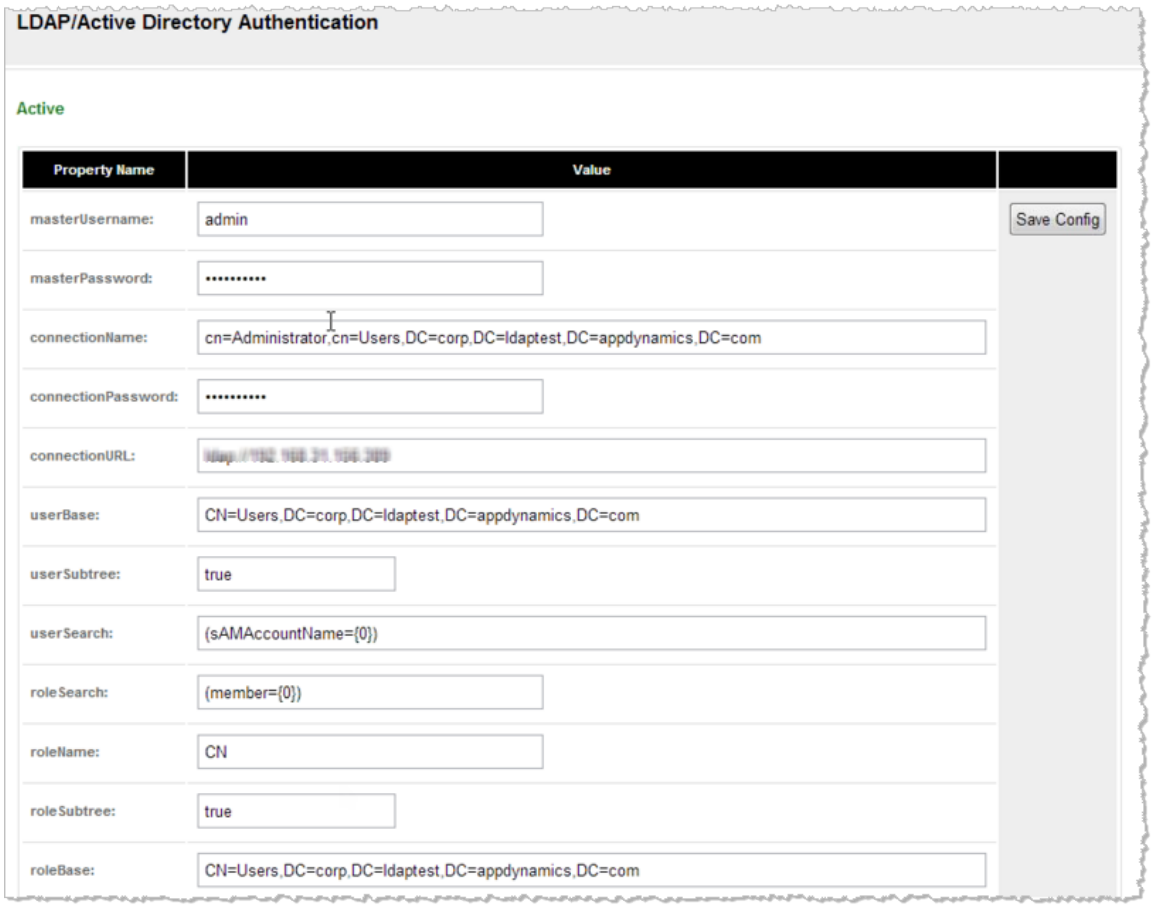


Setup LDAP/Active Directory Integration

When LDAP/Active Directory is integrated, your LDAP and Active Directory users matching the filters defined in this section, will be granted AppD4DB permissions.

1. In a browser, go to the security page. For example, http://<*hostname*>:8090/security.

2. Scroll down the **Security Setup** window, you will see the following sections that you must complete and then click **Save Config** to integrate your LDAP or Active Directory Service server with AppD4DB.



To complete the fields in the LDAP/Active Directory Authentication section

Your LDAP/Active Directory server administrator should provide you with the values you need to complete this section.

The following helps you understand the requirements of each property name field of the LDAP/Active Directory Authentication section:

- masterUsername: The AppD4DB user name of the master user or of an administrator.
- masterPassword: The password for the master user.
- connectionName: The user name AppD4DB uses to log on to the LDAP or Active Directory server.
- connectionPassword: The password for the user AppD4DB uses to log on to the LDAP or Active Directory server.
- connectionURL: The URL of the LDAP or Active Directory server
- userBase: The starting point for the search for users in the LDAP directory tree. This is referred to as the distinguished name. You can specify the search base using the following comma-separated objects, which are not case-sensitive:
    - CN: common name

    - OU: organizational unit
    - O: organization
    - C: country
    - DC: domain
- userSubtree: Set this value to "true" to search through the entire user subtree.
- roleSearch: The filter to use for searching groups.
- roleName: The name of the role.
- roleSubtree: Set this value to "true" to search through the entire role subtree.
- roleBase: The starting point for the search for roles.

Enable Authentication Tracking

To log failed and successful logon attempts, add the following code to the end of <*AppD4DBInstallDir*>\conf\logging.properties.

logging.properties

```
org.apache.catalina.realm.level = ALL
org.apache.catalina.realm.useParentHandlers = true
org.apache.catalina.authenticator.level = ALL
org.apache.catalina.authentical.useParentHandlers = true
```

Monitor Access Attempts

You can check to see who has been successful and unsuccessful attempts to log on to the AppD4DB UI in the catalina.*date*.log file located in <*AppD4DBInstallDir*>\apache-tomcat\logs. The contents of this file can help you determine whether you have correctly configured the LDAP/Active Directory settings; if the user cannot log on, their logon attempts will show in this file.

Start AppDynamics for Databases

Windows

Once installation is complete, the **Main Menu** automatically loads in the default browser.

Alternatively, on the machine where you installed AppD4DB, browse to

```
http://localhost:portNumber
```

where "portNumber" is the port number specified during installation for use by AppD4DB.

You can also connect to the AppD4DB GUI from another networked machine (internet or intranet) by browsing to

```
http://serverName:portNumber
```

where "serverName" is the hostname or IP address of the machine on which AppD4DB is installed and "portNumber" is the port number specified during installation for use by AppD4DB.

Initially, there are no collectors reporting. Next you must add the license and collectors .

- Add a License
- Add a Collector

When AppD4DB starts initially, in the Windows Services (Control Panel -> Sevices), you will see new entries, DBTuna GUI ad DBTuna_DB. Once you have added collectors, you will see an additional service for each collector, DBTuna Agent - *CollectorName*.

Linux

1. Run the ./start.sh script from the installation directory (also referred to as:<appd4db_home>).

2. Then open a URL to:

```
http://serverName:portNumber
```

where "serverName" is the hostname or IP address of the server running AppD4DB
and "portNumber" is the port number specified during installation for use by AppD4DB.

The **Main Menu** opens. Initially, there are no collectors reporting. Next you must add the license and collectors.

- Add a License
- Add a Collector

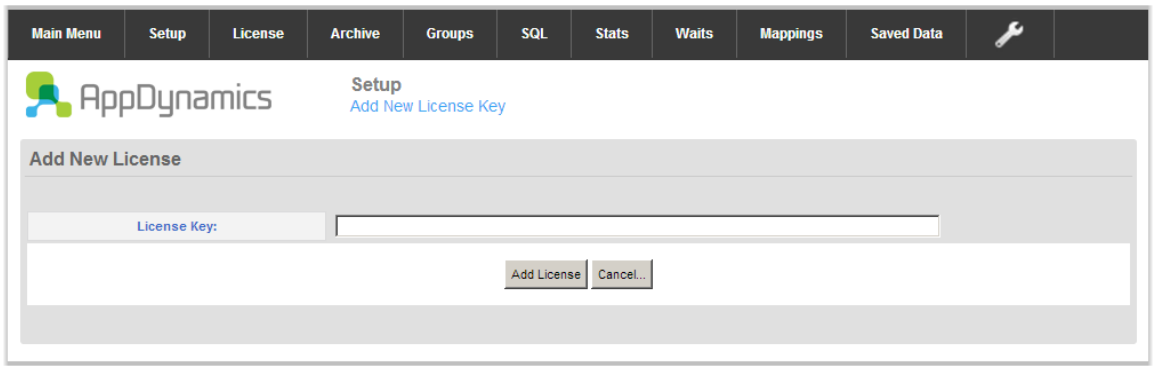
When AppD4DB starts initially, in the processes list (cmdshell: ps), you will see new entries, DBTuna GUI ad DBTuna_DB. Once you have added collectors, you will see an additional service for each collector, DBTuna Agent - *CollectorName*.

Add a License

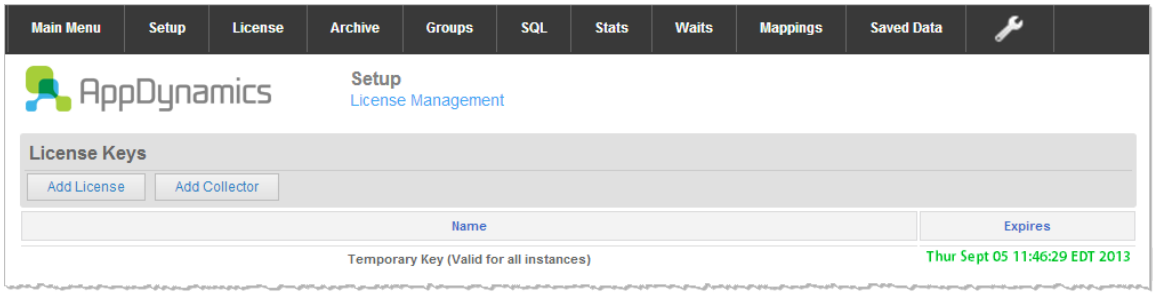
Once you have purchased or applied for a trial of AppDynamics for Databases (AppD4DB), you will receive an email from AppDynamics Support with your license key. Your license key is tied to the number of databases, and the number of .NetApp and server controllers that you want to monitor. At the end of the installation, when the AppD4DB UI runs for the first time, you are prompted to enter your license key. To apply your key:

1. Click **Setup -> Add License**.

2. Copy your license key from the email AppDynamics Support sent you, paste it in the **License Key** box, and then click **Add License**.



The **License Management** window displays the added license.

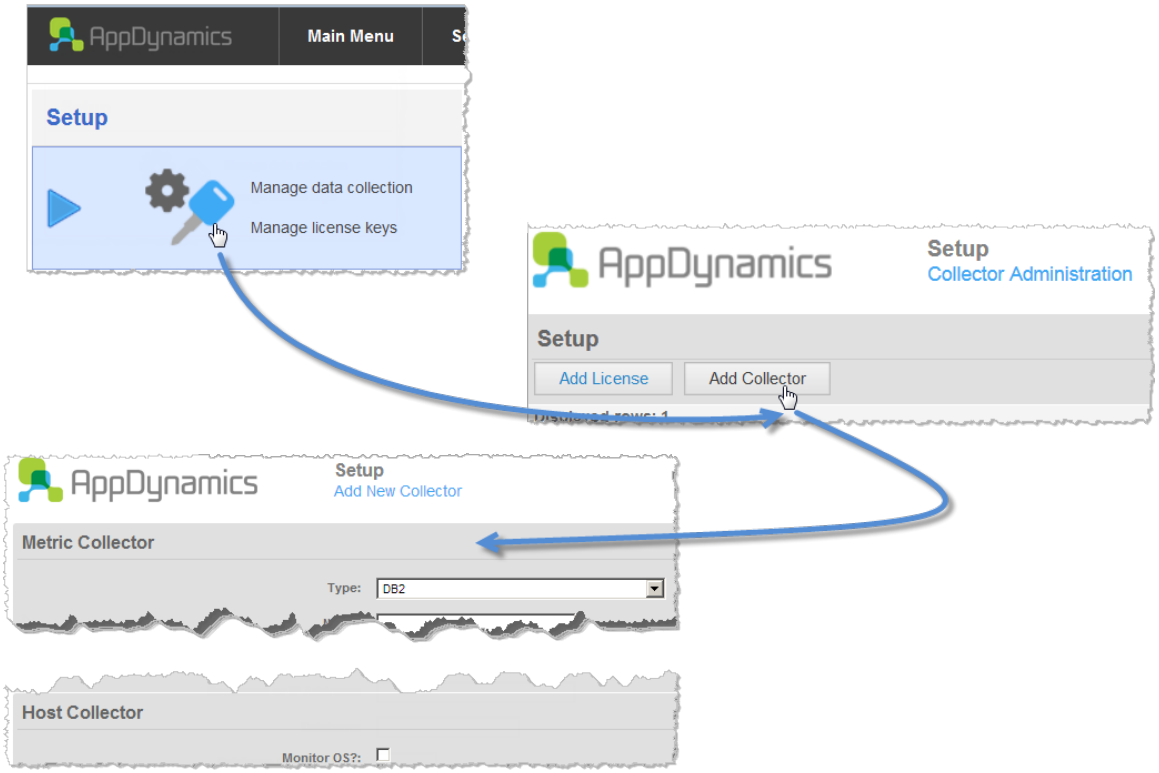


Add a Collector

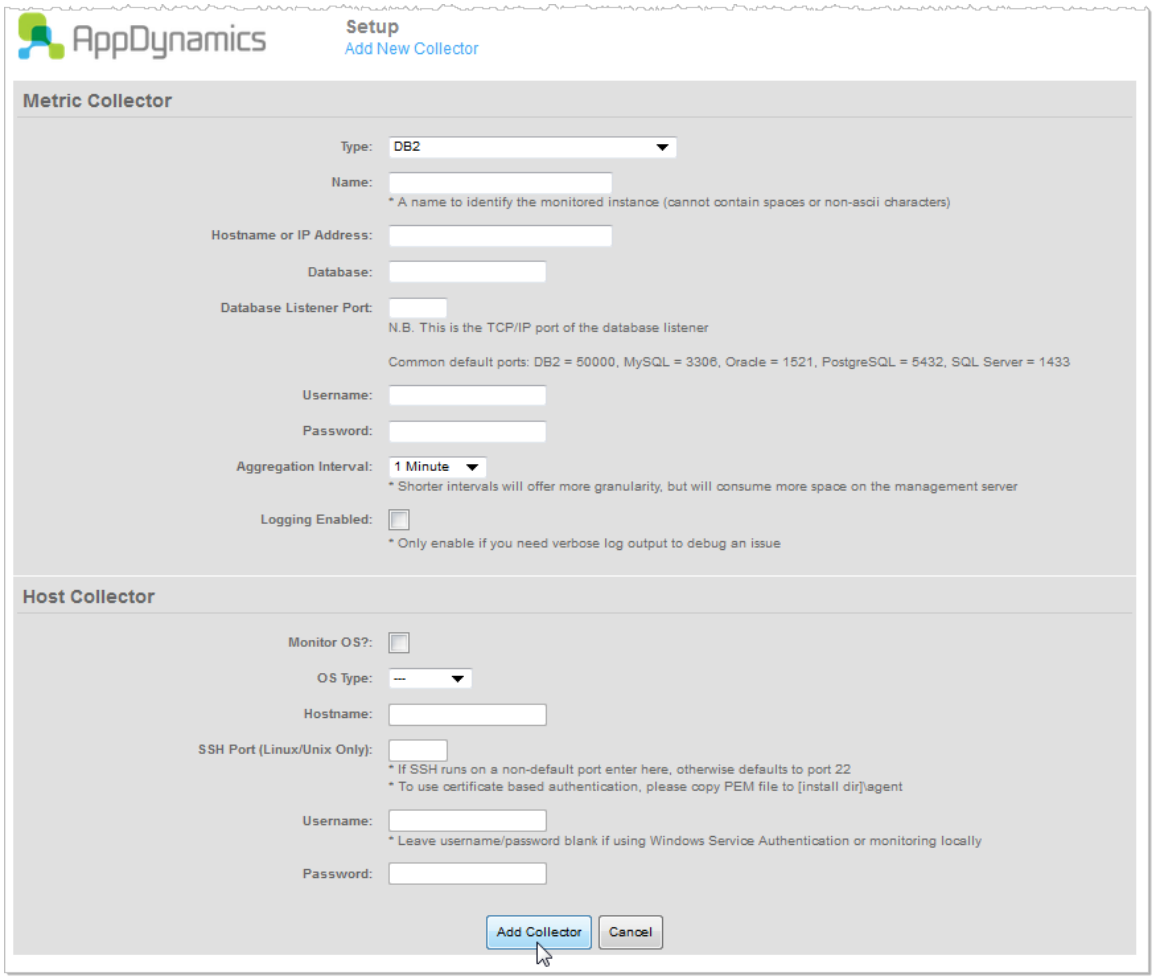
You can create collectors that run on the AppD4DB server to monitor any of the following systems:

- **Databases**: DB2, MySQL, Oracle, PostgreSQL, Sybase ASE, and Sybase IQ
- **Servers**: Linux Server, Solaris Server, or Windows Server
- **Storage**: NetApp

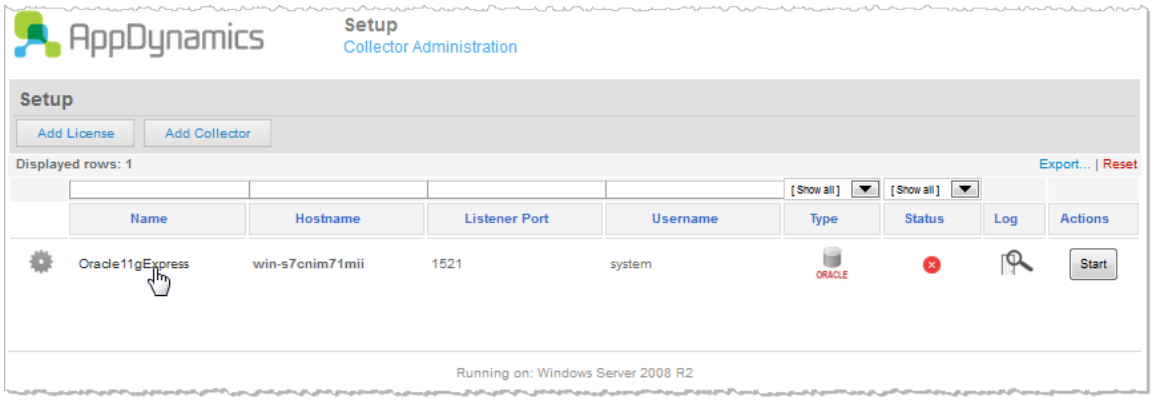
1. Click **Main Menu -> Setup -> Add Collector**.



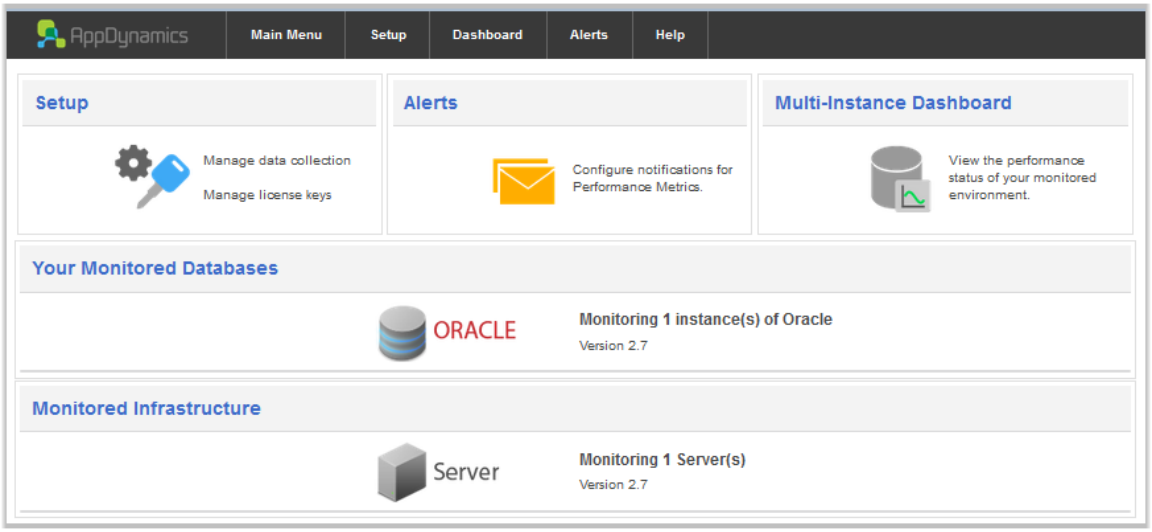
2. Complete each field of the **Add New Collector** window and then click **Add Collector**.



3. After you add a collector, the new collector appears on the **Collector Administration** window.



You will also now see your collectors on the Main Menu.



To change any of the collector attributes, click its name in the **Name** column of the **Collector Administration** window. The **Edit Collector Configuration** window appears where you can change the collector attributes except the **Type** and **Name** attributes.

Use the following to help you complete the fields of the **Add New Collector/Edit Collector Configuration** window. The fields that display depend on the type of collector you selected.

Complete the Metric Collector Section

Type: Select the collector type. There are three categories of collectors and collector types within these categories:

- **Database Collectors**: DB2, MySQL, Oracle, PostgreSQL, and Sybase ASE
- **Server Collectors**: Linux Server, Solaris Server, or Windows Server
- **Storage Collectors**: NetApp

Once you select the Type, the fields displayed may change because the information required differs between collector types. The type appears in the **Type** column of the **Collector Administration** window.

Name: Specify a name to uniquely identify the monitored instance. The name cannot contain spaces or non-ASCII characters. This name appears in the **Name** column of the **Collector Administration** window.

Hostname or IP Address: Specify the hostname or IP address of the machine serving the database. The hostname or IP address appears in the **Hostname** column of the **Collector Administration** window.

Database: Specify the database name, named instance or SID of the database to monitor.

Failover Partner: (Microsoft SQL Server) If you use Database Mirroring, enter the hostname of the IP address of the Failover Partner here.

EnterpriseDB?: (Postgre) Click if your Postrgre database installation is an EnterpriseDB distribution.

Connect as a sysdba: (Oracle) Click if you want to connect as user sys and enable password files. Running the collector using a sysdba account allows the collector to monitor an Oracle instance in standby mode, an instance that is used for failover which is being replicated from the main active instance.

SID or SERVICE_NAME: (Oracle) Specify the Oracle System ID (SID) or service name that uniquely identifies the database the collector monitors.

Database Listener Port: Specify the TCP/IP address of the port the database uses for communicating with the AppDynamics for Database (App4DB) controller. The database list port appears in the **Listener Port** column of the **Collector Administration** window.

Username: Specify the name of the user AppD4DB uses to connect to the database. This user must have the Required Database Permissions

Password: Specify the password of the user AppD4DB uses to connect to the database.

Aggregation Interval: Specify the time interval at which AppD4DB should write out the performance data to the repository. Shorter aggregation intervals provide more granularity and closer to real-time access to data; however, more disk space is required to store historical data and there will be a small increase in overhead as the target system is queried more frequently. For more information, see Configure the Aggregation Interval.

Logging Enabled: Click to enable verbose mode logging, which logs all communications between the controller and the collector. Enable only during troubleshooting because logging can consume a lot of disk space. If you have enabled logging, you can click the logging icon in the **Log** column of the **Collector Administration** window to view the log file. The log files are located in the <App4DBInstalDir>\agent directory and have the format *CollectorName*_out.log and *Collector Nam_err.log*.

Complete the Host Collector Section

Monitor OS?: Click if you want CPU consumption metrics collected from the monitored host. For more information, see Infrastructure Monitoring (Monitor OS?).

OS Type: Specify the operating system of the monitored host: Linux, Solaris, or Windows.

Hostname: Specify the hostname or IP address of the monitored host.

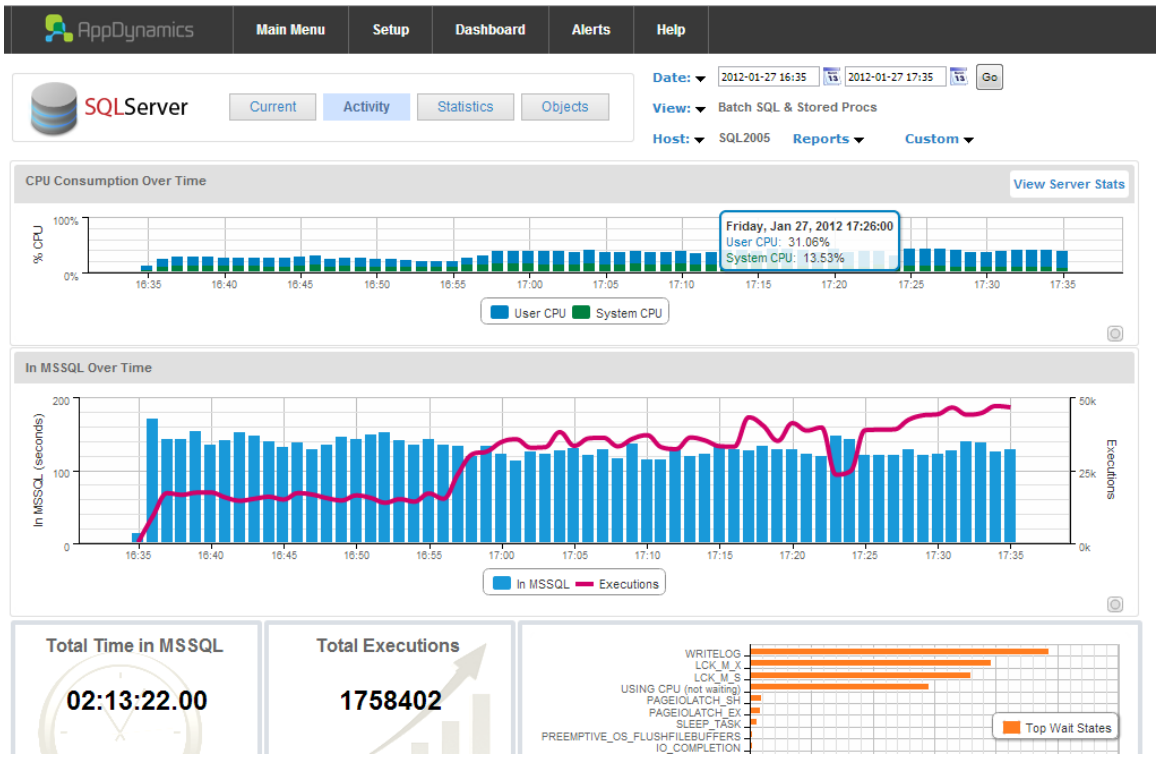
SSH Port (Linux/Unix Only): Specify the Secure Shell (SSH) port number AppD4DB should use for encrypted communications with the monitored host. The default port number of 22 will be used if you do not specify a different port number here. AppD4DB also supports certificate-based authentication via Privacy Enhanced Mail (PEM). To implement certificate-based authentication, copy the PEM file to the <App4DBInstallDir>\agent directory.

Username: Specify the name of the user AppD4DB uses to log on to the monitored host

Password: Specify the password of the user AppD4DB uses to log on to the monitored host.

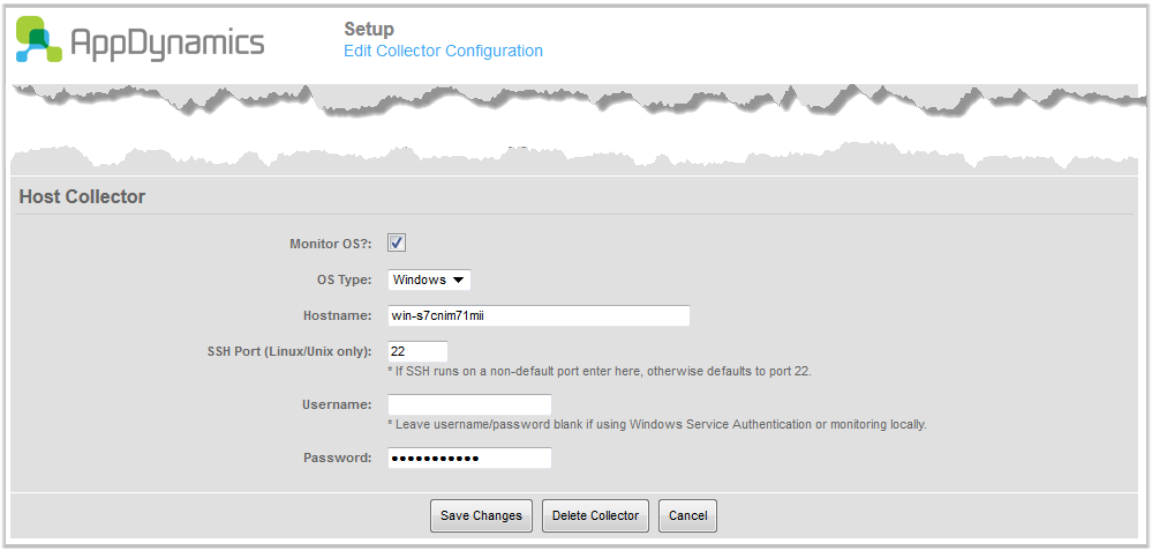
Infrastructure Monitoring (Monitor OS?)

AppDynamics for Databases (AppD4DB) can be configured to collect CPU consumption metrics from the monitored host. An example of these metrics can be seen in the **CPU Consumption Over Time** chart as shown below.



- Collecting Information from Linux and Solaris
- Collecting Information from Windows
- Using Windows Authentication to Monitor SQL Server
- NetApp Permissions

Note: To enable the Host Collector, on the **Add Collector** or **Edit Collector Configuration** pages you must click the **Monitor OS?** checkbox and enter details of **OS Type** and **Hostname** (or IP address.)



Collecting Information from Linux and Solaris

AppDynamics for Databases can monitor Linux or Solaris from any of its supported platforms.

To collect OS metrics from Linux or Solaris, the AppD4DB Host Collector makes an ssh connection to the monitored host to gather information using standard commands. For the AppD4DB collector to successfully collect information, the monitored host must support ssh connections, and the AppD4DB Host Collector must have a hostname, username and password defined.

Collecting Information from Windows

AppD4DB can only monitor Windows from a Windows platform.

The AppD4DB Host Collector will collect OS metrics from a Windows host via Windows Management Instrumentation (WMI). To collect OS metrics from a remote Windows machine, the AppD4DB Host Collector must be an authenticated user; therefore, either the username and password should be defined explicitly in the OS Monitor settings, or the AppD4DB Windows service should be modified to run as an authenticated user, in the same way as outlined in Using Windows Authentication to Monitor SQL Server.

Using Windows Authentication to Monitor SQL Server

1. When creating a collector on the **Collector Setup** page, do not specify a username and password.

2. Configure the AppD4DB windows service to log on as the desired windows account with SQL Server access.

a. Go to the Windows Services UI from Control Panel. Select the AppDynamics for Databases collector service and right click to display the properties.

b. Click **Log On** and then enter the credentials for the user.

c. Click **OK**, and then try starting the collector either from the services panel or from the **Collector Administration** page

3. Repeat the above process for the AppDynamics for Databases GUI Windows Services.

The AppDynamics for Databases GUI windows service also needs to be configured to log on as the desired windows account with SQL Server access; this is because to display some windows within AppDynamics for Databases, such as the Current window and the Objects window, the UI process needs to make a direct connection to the monitored SQL Server to request information.

If you are running AppDynamics for Databases on Linux then you must use SQL Server authentication.

NetApp Permissions

For complete AppDynamics for Databases functionality in a NetApp environment, the NetApp Metric Collector needs access NetApp API access.

Here is an example user creation script if you would like to create a new dedicated user for AppDynamics for Databases with relevant privileges:

```
useradmin role add Appd4db_Role a api*,login-http-admin
useradmin group add Appd4db_Group -r Appd4db_Role
useradmin user add appd4db -g Appd4db_Group
```

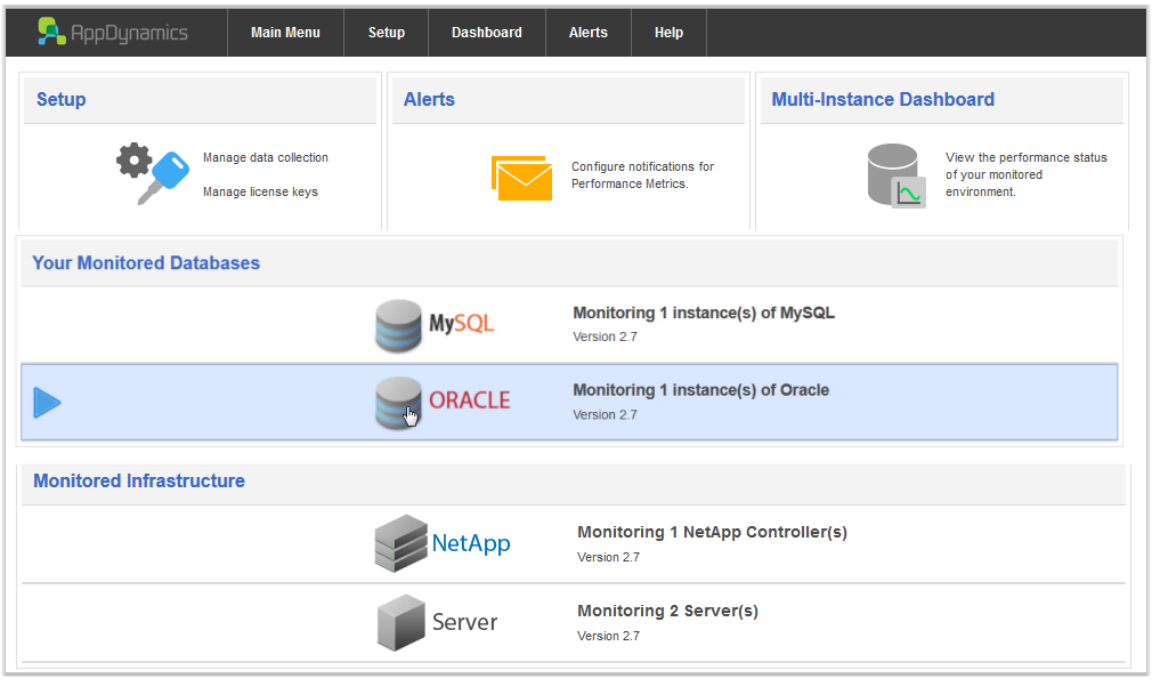
AppDynamics for Databases can use either HTTP or HTTPS (i.e. SSL) to access the performance data. If your filer uses SSL, then just check the SSL checkbox on the Add New Collector or Edit Collector Configuration page.

Use the Database Platform Window

- Accessing the Database Platform Window
- Setting the Time Range
  - Setting a Relative Time Period
  - Setting an Absolute Time Range
- Viewing Historical Performance Data (Activity tab)
  - Viewing Historical %CPU Consumption
  - Viewing Time Spent Running SQL and Procedures and Database Throughput
    - SQL Details
    - SQL Explain Plan
- Viewing Up-to-the-Second Performance (Current tab)
- Database Statistics
- Database Objects

Accessing the Database Platform Window

From the **Main Menu** window, click a database icon to open the database platform window for that particular database.



The **Activity** window appears.

Setting the Time Range

For the **Activity** and **Statistics** tabs, you can select the time period using the **Date** list and calendar settings at the top right of the window.



Setting a Relative Time Period

Use the **Date** list to select a relative time period; options range from “Last 5 Minutes” to “Last 2 Weeks." If you select a time period of less than 30 minutes, then the UI will auto-refresh and retrieve the latest historical data at the end of the collector aggregation interval.

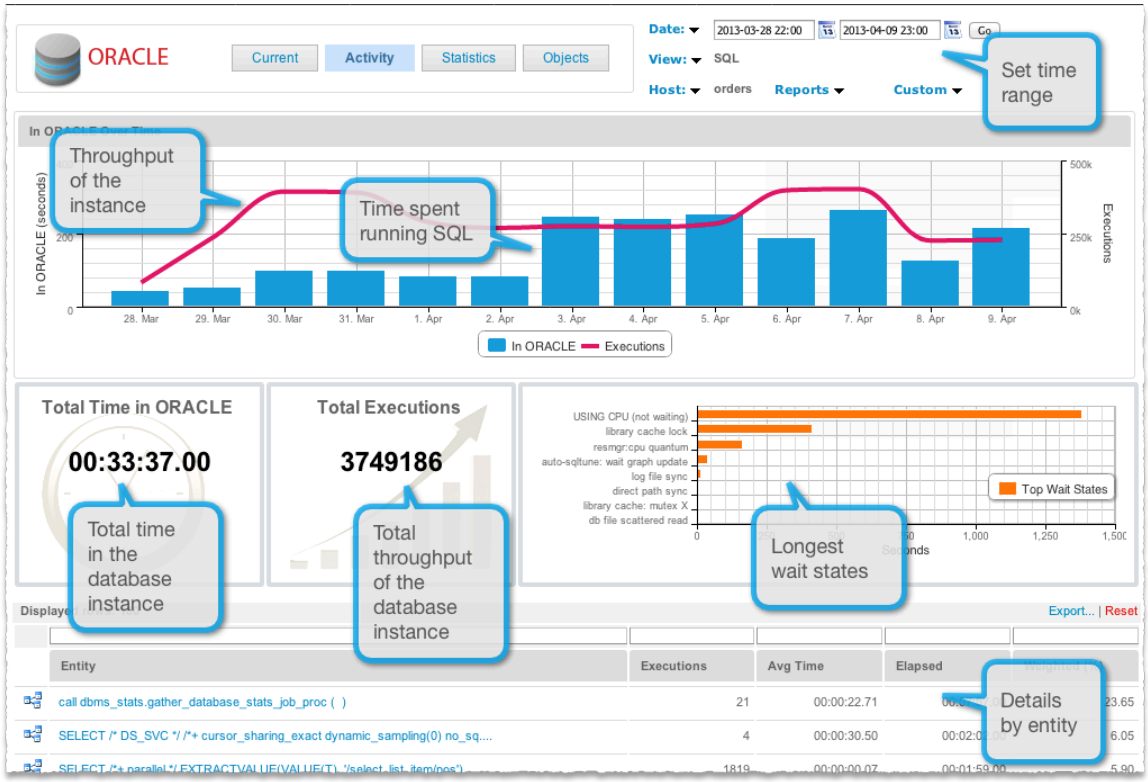
Setting an Absolute Time Range

Use the calendar controls to select an absolute time range. Set the dates (in the form “YYYY-MM-DD HH:MI:SS”) and click **Go** to display data for this range.

Viewing Historical Performance Data (Activity tab)

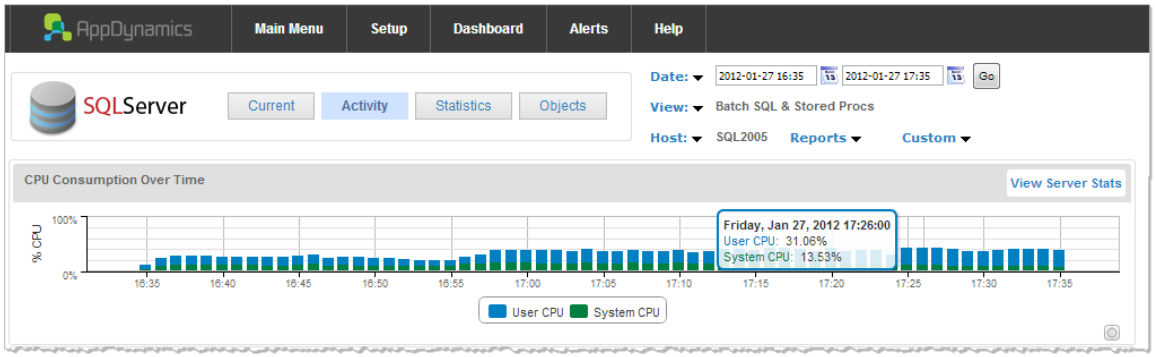
The AppDynamics for Databases **Activity** window is the default tab of the Database Platform window. Most monitoring activities occur here. The **Activity** tab displays historical performance

data that is retrieved from the repository.



Viewing Historical %CPU Consumption

If, when you configured the AppDynamics for Databases collector, you enabled the **Host Collector** and entered valid credentials, then you will see the historical %CPU consumption of the server in a graph at the top of the **Activity** tab.



The **CPU Consumption Over Time** graph displays the percentage CPU consumption of the physical or virtual server running the monitored database instance. The figure displayed is a percentage of all CPUs in the machine, and for all processes running, not just the database process.

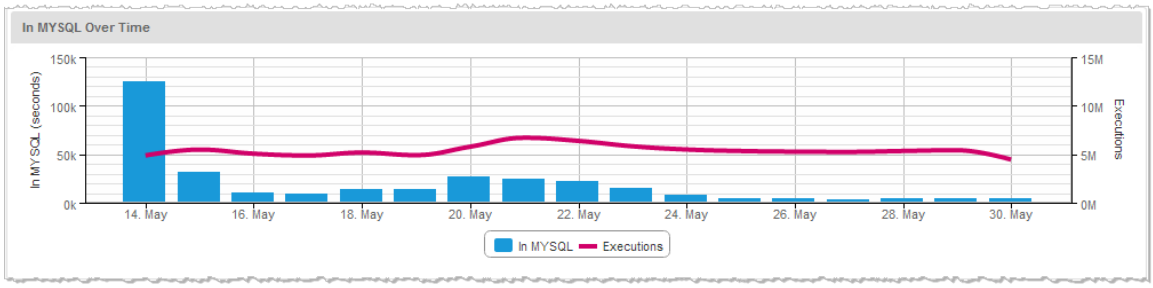
The percentage figure is broken down by:

- **User CPU**: CPU consumed by application processes.
- **System CPU**: CPU consumed typically by the OS, often for paging/swapping or background kernel operations.
- **I/O Wait**: CPU time consumed while the CPU waited for retrieval of data from the I/O subsystem (only available for Unix/Linux platforms).

Viewing Time Spent Running SQL and Procedures and Database Throughput

The In "Database" Over Time graph on the **Activity** tab is a very powerful graphical view of performance. It displays two series of data:

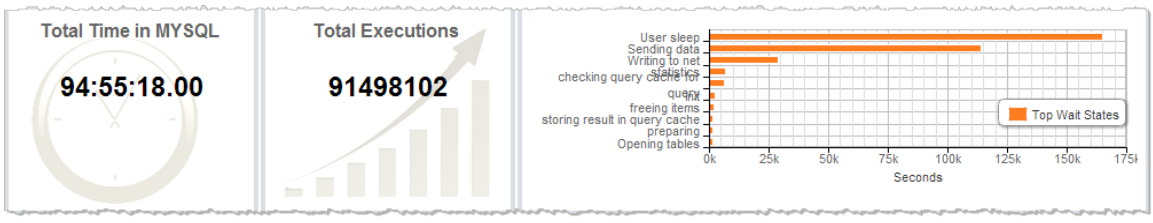
- The time spent running SQL and procedures within the database instance.
- The throughput of the database instance, measured as a count of the SQL executions.



Instance Level Summary Data

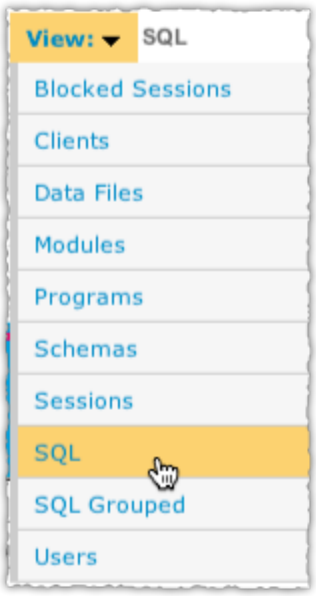
Three boxes on the **Activity** tab display summaries of:

- **Total Time in Database**: The total time spent running SQL and procedures within the database instance. Expressed in hours, minutes and seconds (HH:MI:SS.XX).
- **Total Executions**: A count of all SQL and Stored Procedure executions processed over the selected time period.
- **Top Wait States**: A graphical summary of the top database wait states for the selected time period based on seconds spent waiting.



Database Activity Detail Data

The tabular data displayed at the top of the **Activity** tab shows an ordered list of the top entities as selected in the **View** menu.



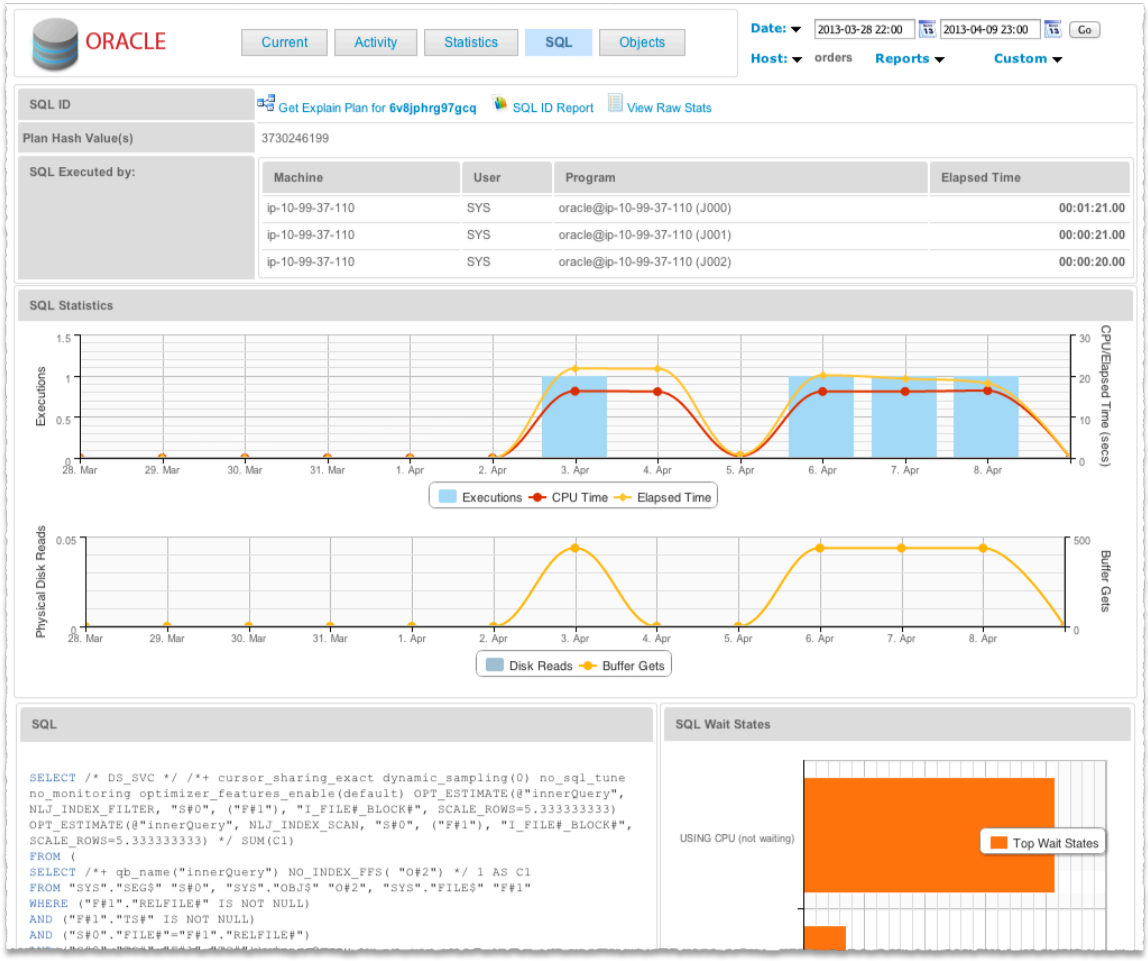
The default view displays the SQL. The number of entities that can be viewed vary by database platform, but commonly contain Top Users, Top Client (machines), Top Sessions, etc.

You can filter the list using the text fields at the top of the list.



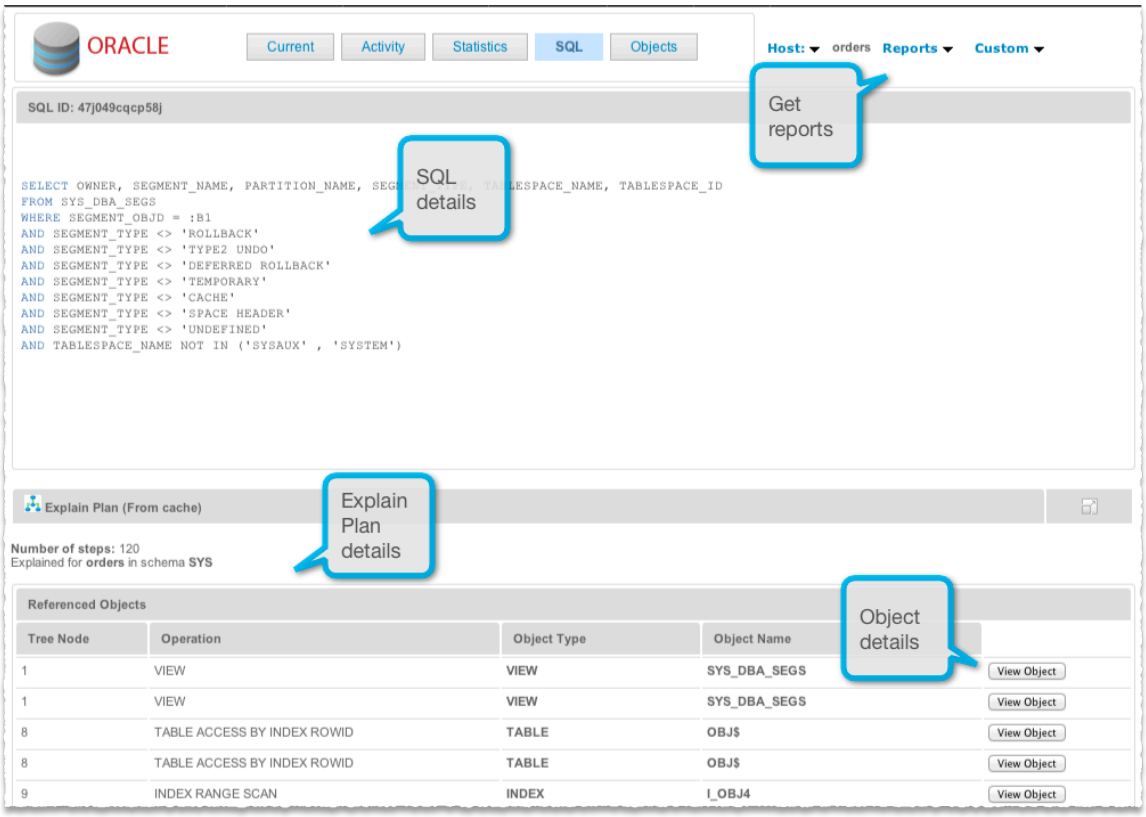
SQL Details

In the Activity Detail Data list click a SQL Statement to open the SQL Details window.



SQL Explain Plan

In the Activity Detail Window or in the SQL Details window, click the Explain Plan icon to see the statement and its Explain Plan.

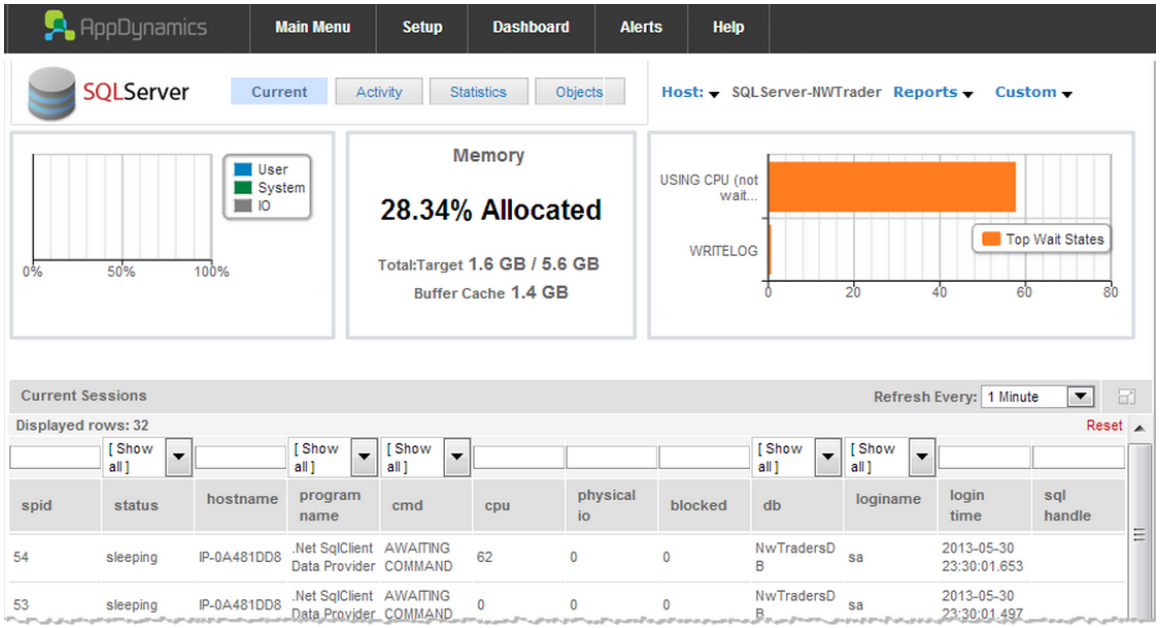


Viewing Up-to-the-Second Performance (Current tab)

Click **Current**.

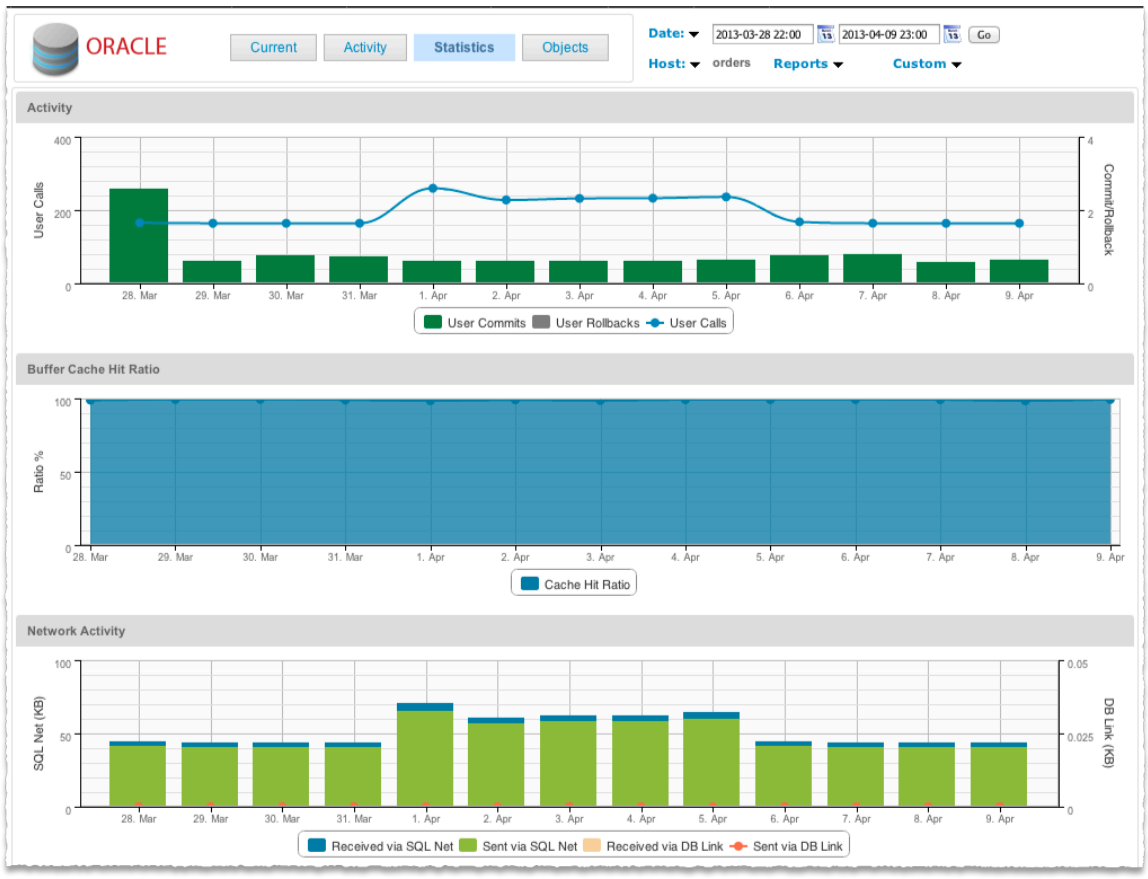
The AppDynamics for Databases **Current** tab is the workspace where you can see an up-to-the-second view of performance. Rather than reading from the historical repository, the **Current** tab connects directly to the monitored instance and requests current information using SQL queries.

The appearance of the **Current** tab may vary somewhat depending on the database platform type; however, it usually displays a graphical summary of current key performance indicators and a list of all currently connected database sessions or processes.



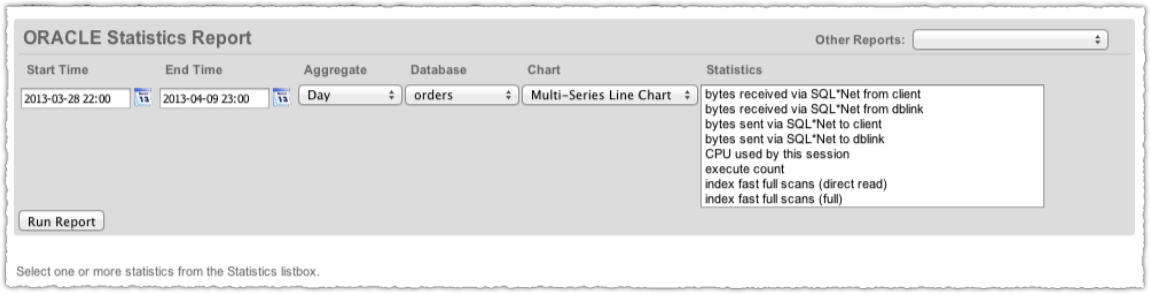
Database Statistics

The Statistics window displays historical information about key performance indicators of the database instance.

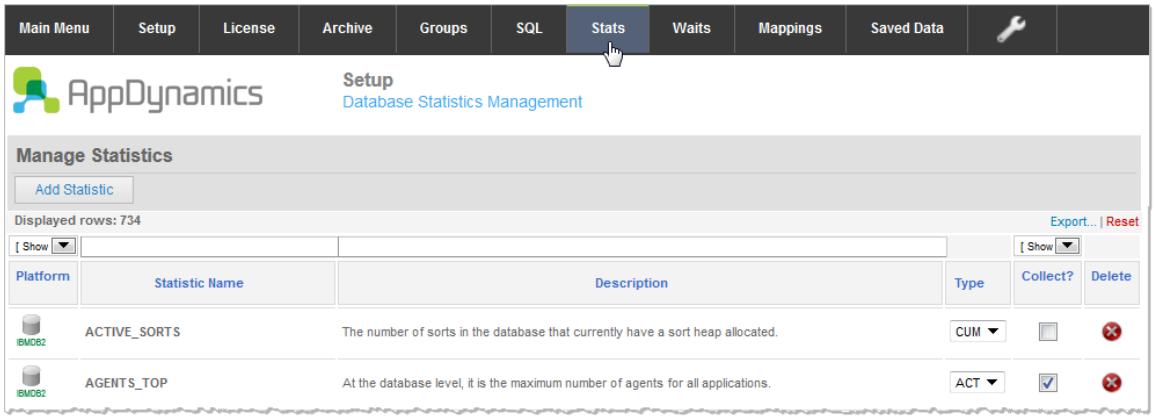


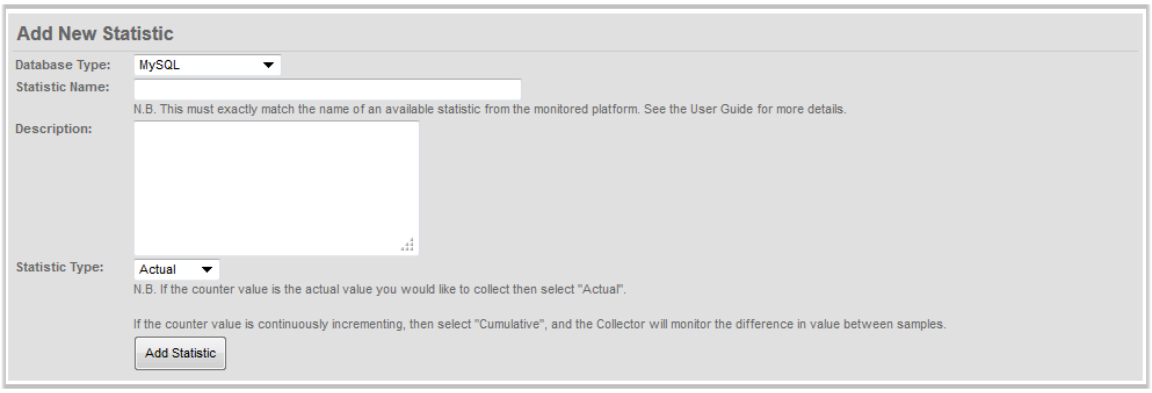


To display historical graphs on any other collected statistic, use **Reports -> Database Statistics Report**.



You can configure AppDynamics for Databases to collect tens or even hundreds of individual statistics. Add statistics using the **Setup -> Stats** window.



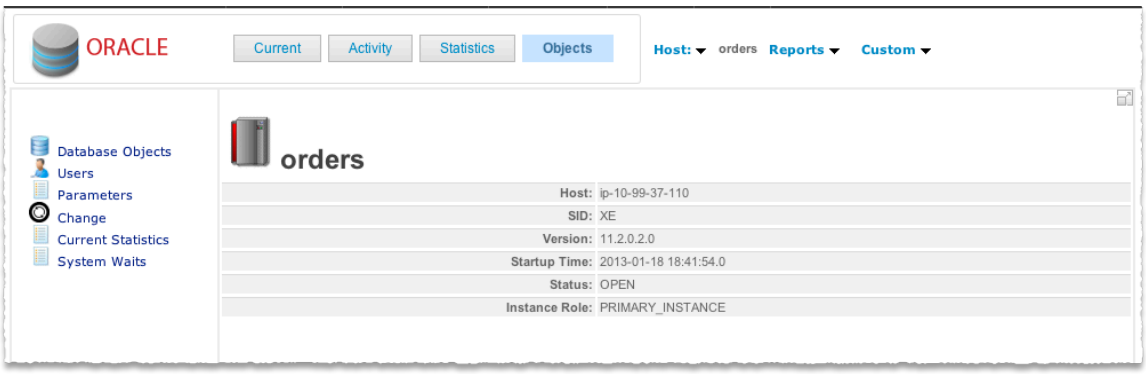


The following table lists the source of statistical data. Any new statistic name must match exactly to the returned value from the source.

Database	
DB2	*additional statistics not currently supported for this platform*
MySQL	show status
SQL Server 2000	master..sysperfinfo
SQL Server 2005 and above	sys.dm_os_performance_counters
Oracle	v$sysstat
PostgreSQL	*additional statistics not currently supported for this platform*
Sybase ASE	master.dbo.sysmonitors

Database Objects

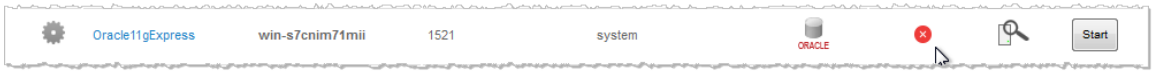
The Objects window contains links to meta-data about the database configuration and schema by connecting directly to the monitored instance and requesting current information using SQL queries.



The Objects window also contains information about configuration changes within the database instance since the collector began monitoring.

Verify Collector Setup

The collector is now ready to start collecting performance data! Go back to the **Setup** window and click the **Start** button for each collector.



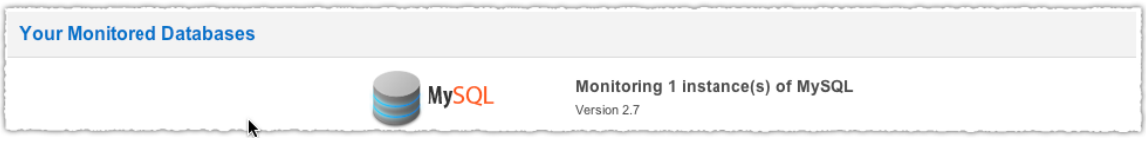
When successfully started, the collector **Status** column displays a green tick symbol.



Congratulations you have set up your first collector!

Once the collector is up and running, in just a short time you can start viewing the historical activity data. By default, AppDynamics for Databases (AppD4DB) logs data every one minute. You configure this aggregation interval on the **Add New Collector** and **Edit Collector Configuration** windows. For more information see, Add a Collector.

The **Main Menu** window now has an collector icon you can click to view the activity for the collector.



Click the collector icon to open the **Activity** window and see what AppDynamics for Databases has captured for you!

For information on using and interpreting the collector windows, see Use AppDynamics for Databases.

Troubleshooting Collector Startup

If the collector does not start and the icon in the **Status** column displays a red cross, then you can view the contents of the collector log file to investigate the problem. Click the **Log** file icon for the collector on the **Collector Administration** window. The log files are located in the <App4DBInstallDir>\agent directory and have the format *CollectorName*_out.log.

For assistance, please contact AppDynamics Support.

Integrate AppD4DB with AppDynamics Pro

Configuring AppDynamics Pro to interface with AppDynamics for Databases (AppD4DB) differs depending on the version of AppDynamics Pro you are using. See the configuration topic that

applies to your installation.

- Integrate AppD4DB with AppDynamics Pro 3.7
- Integrate AppD4DB with AppDynamics Pro 3.6

If you are installing AppD4DB on the same host as the AppDynamics Controller, see also Installing AppD4DB and AppD Controller on the Same Host.

See also, Troubleshooting.

Integrate AppD4DB with AppDynamics Pro 3.7

- Prerequisites for AppDynamics for Databases Integration
- Configuring AppDynamics to Interface with AppDynamics for Databases
    - To Configure AppDynamics to Interface with AppDynamics for Databases

This topic describes how you can you link to AppDynamics for Databases (AppD4DB) from a monitored database, server, or storage system (NetApp) that is discovered by AppDynamics and supported by AppD4DB. This integration provides access to the deep database performance metrics provided by AppD4DB.

Prerequisites for AppDynamics for Databases Integration

To use this integration you must have an AppD4DB license.

AppDynamics for Databases must be configured to monitor the databases that you want to link to from AppDynamics.

In addition, if you are installing AppD4DB and the AppDynamics controller on the same host, you may need to change the port number AppD4DB uses because the default port number for both products is 8090. For information about changing the port number on AppD4DB, see Installing AppD4DB and AppD Controller on the Same Host.

See the AppDynamics for Databases documentation for information about using AppD4DB.

Configuring AppDynamics to Interface with AppDynamics for Databases

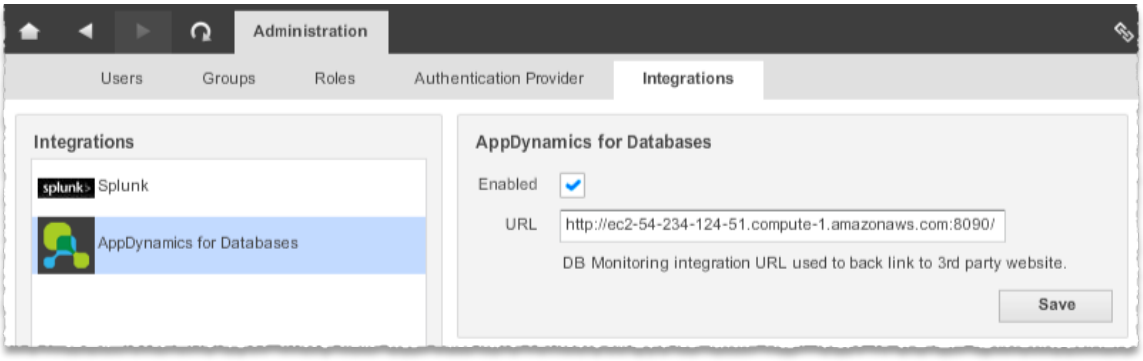
To Configure AppDynamics to Interface with AppDynamics for Databases

1. Click the Setup menu in the upper right section of the screen.

2. From the drop-down menu, click **Administration**.

3. Click the **Integration** tab.

4. Click AppDynamics for Databases in the Integrations list.



5. Check the **Enabled** check box to enable the integration.

6. Enter the URL of your AppDynamics for Databases installation. The trailing slash is required.

7. Click **Save**.

Integrate AppD4DB with AppDynamics Pro 3.6

Prerequisite:

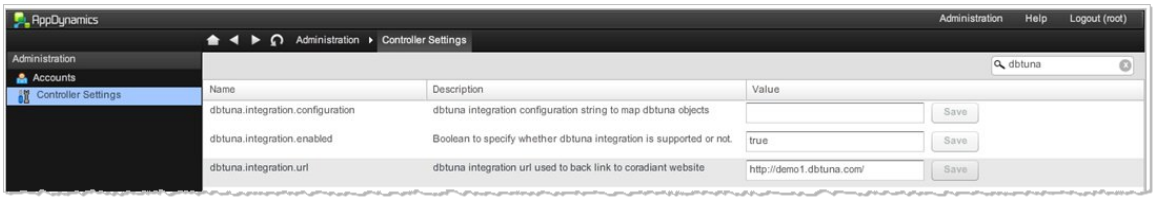
If you are installing AppD4DB and the AppDynamics controller on the same host, you may need to change the port number AppD4DB uses because the default port number for both products is 8090. For information about changing the port number on AppD4DB, see Installing AppD4DB and AppD Controller on the Same Host.

To Configure AppDynamics Pro 3.6 to Interface with AppDynamics for Databases

1. Go to http://server:port/controller/admin.html.

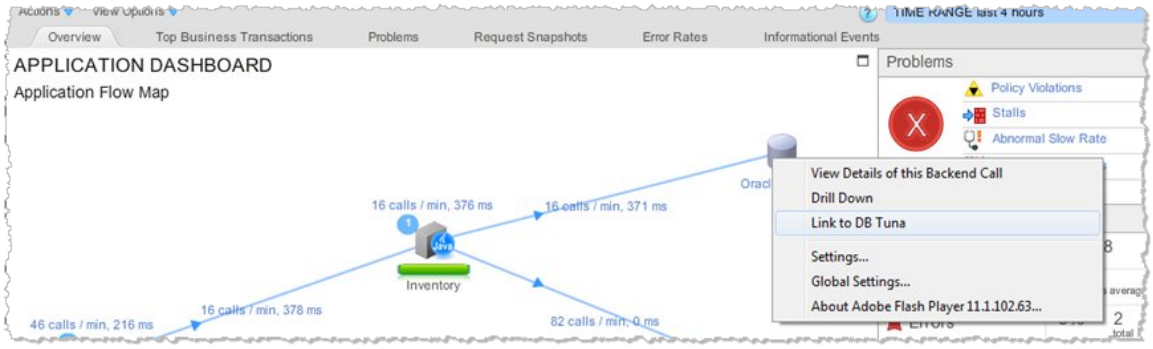
2. Log in as root and select **Controller Settings.** You can filter on dbtuna in the box at the top.

3. Set dbtuna.integration.enabled to "true" and dbtuna.integration.url to "http://demo1.dbtuna.com/".

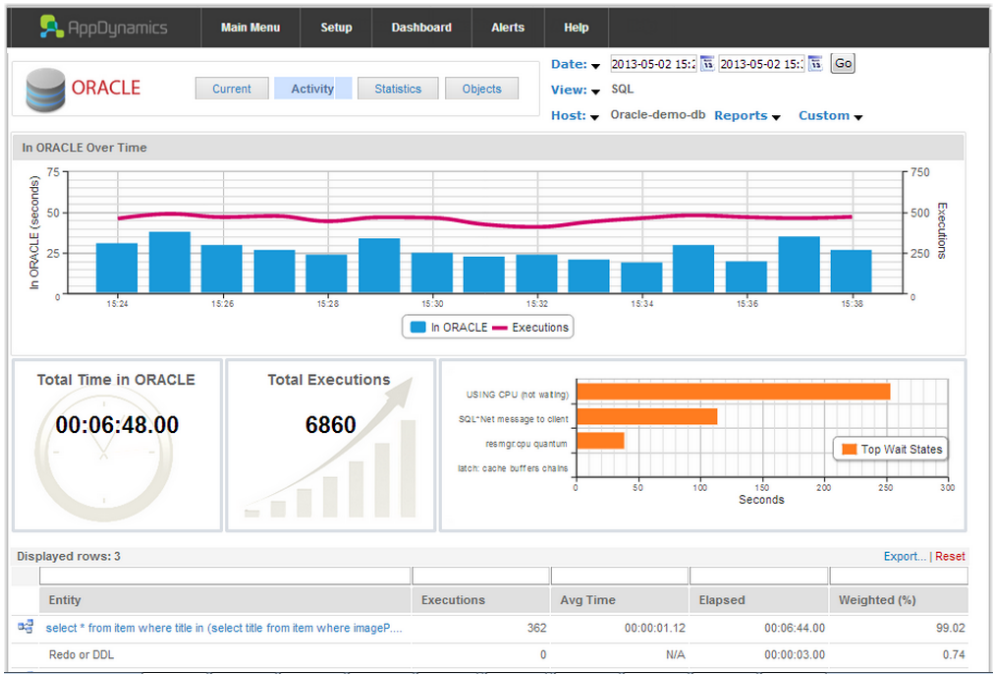


4. Login to the DBTuna UI Application Dashboard.

5. On the Flow View, right-click a database backend and select **Link to DB Tuna**. This can be done anywhere you see a database on the Application Dashboard, BT Dashboard, Tier Dashboard, Node Dashboard, and within Snapshots.



6. AppDynamics for Database displays the activity for the database selected.



Installing AppD4DB and AppD Controller on the Same Host

When installing AppD4DB on the same host as the AppDynamics Controller, there is a port conflict. Both products use port 8090 for the UI by default. That means that the AppD4DB UI server will fail to start after the install. You must change the port number and then restart the server.

To Change the Port Number of the AppD4DB UI Server:

1. Open the following file in a text editor:
   AppD4DBInstallDir/apache-tomcat/conf/server.xml where "AppD4DBInstallDir" is the directory where AppD4DB is installed.

2. Find the only line containing 8090 and change the number to the number of the desired port.

   Note: Do not use 8091 because this port is used for the MySQL connection between AppD4DB

and the MySQL server database where it stores the captured data.

3. Restart the AppD4DB process as follows:

Windows: Restart the Windows service named: "DBTuna." Its description in the Task Manager is "DBTuna GUI."

Linux: From the AppD4DB home directory, rRun ./stop.sh followed by ./start.sh

Use AppDynamics Pro with AppDynamics for Databases

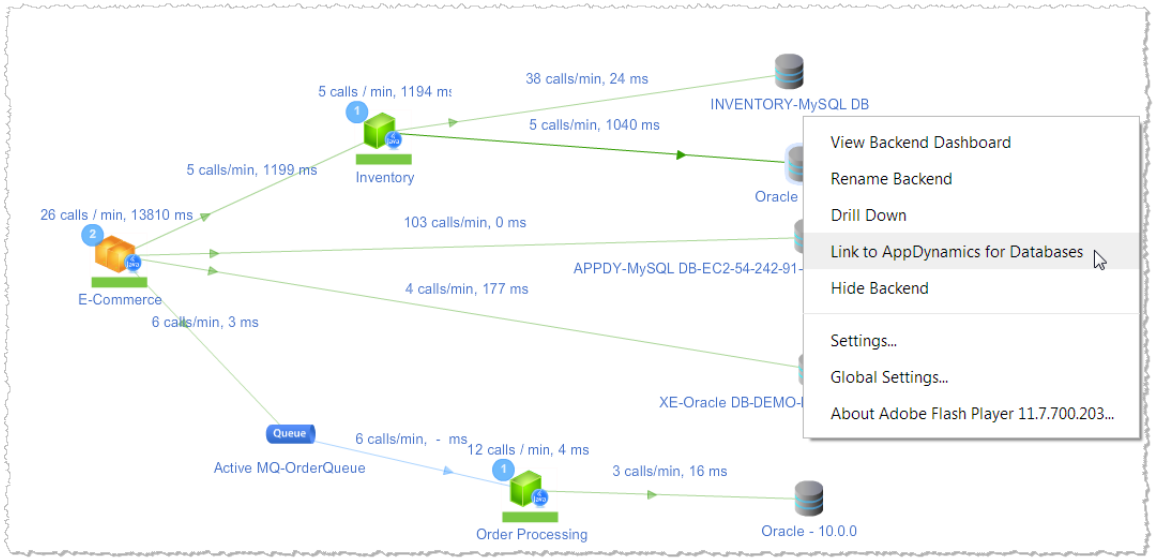
The following topics introduce you to linking to and using AppDynamics for Databases from within AppDynamics Pro. For more detailed information, see
Use AppDynamics for Databases

- Drilling Down from AppDynamics
- Main Window of AppDynamics for Databases

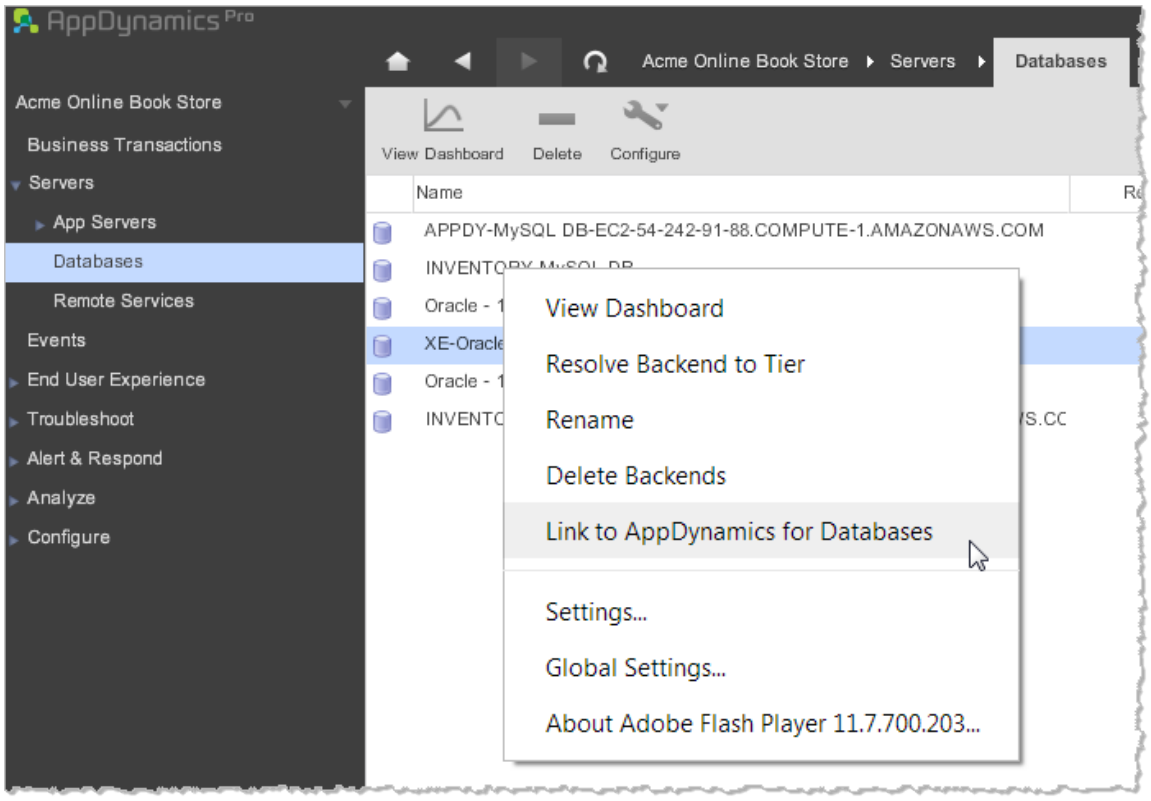
Drilling Down from AppDynamics

Once you have configured AppDynamics and AppDynamics for Databases integration, from within AppDynamics you can drill down into database calls in AppDynamics for Databases as follows:

Right-click or CTRL-click a flow map database icon.



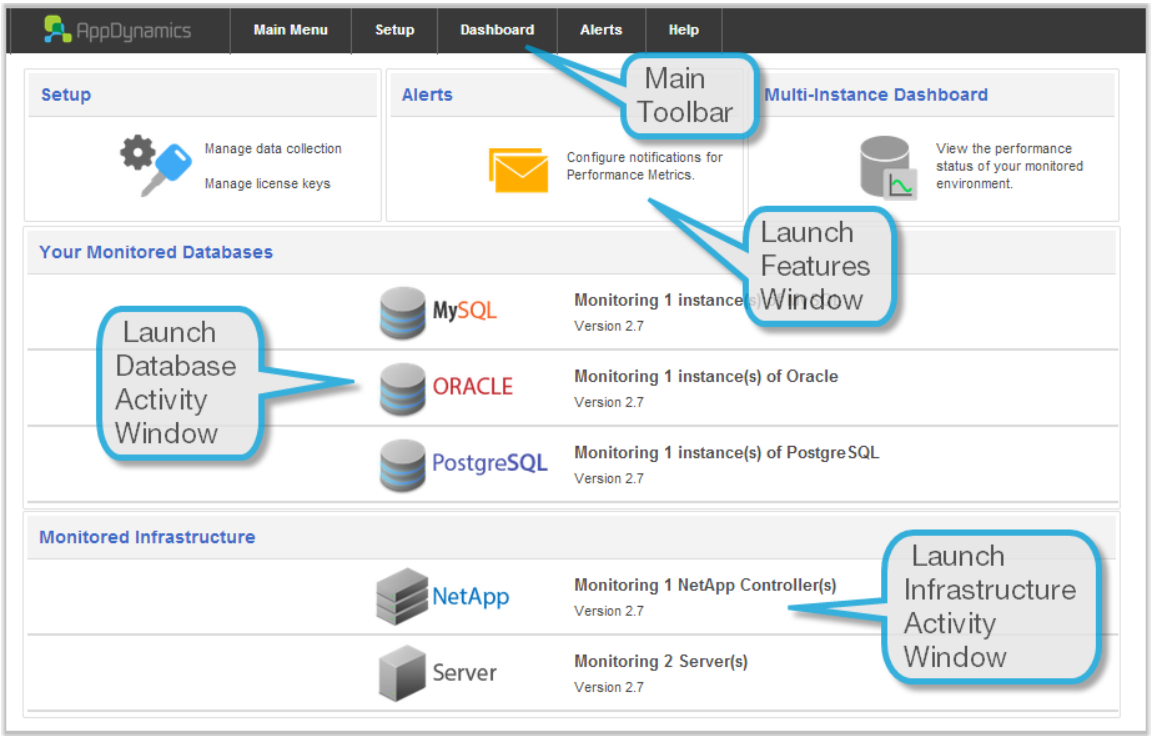
On the **Databases** dashboard, right-click a database for which you have setup an AppD4DB collector and click **Link to AppDynamics for Databases**.



The **Action** menu option is also available in the **Slowest Database Calls** tab.

Main Window of AppDynamics for Databases

When you link to AppDynamics for Databases, the Main Window appears.



The Main Menu window links to all areas within the product:

- The toolbar is common across all windows and accesses the Main Menu, Setup, Dashboard, Alerts, and Help windows.
- Quick buttons access the Setup, Email Alerts, and the Multi-Instance Dashboard.
- The Monitored Databases section lists all the databases being monitored by vendor. Each row shows how many instances of the database are being monitored and the version. The rows access the Activity windows for the specific database technologies.
- The Monitored Infrastructure section lists the NetApps, if any, and underlying Servers. The rows access the Activity windows for each infrastructure resource.
- The AppDynamics for Databases build number is at the bottom of the window. You may need this if you need to contact technical support.