



---

# Developing Dynamic Applications for Database Monitoring

SL1 version 12.3.0

---

# Table of Contents

<b>Introduction to Database Dynamic Applications</b> .....	<b>4</b>
Prerequisites .....	5
What is a Database Dynamic Application? .....	5
How Do I Allow SL1 to Access the Database? .....	6
What Can I Monitor with a Database Dynamic Application? .....	6
Can I Create My Own Database Dynamic Applications? .....	6
Elements of a Dynamic Application for a Database .....	6
<b>Defining Collection Objects</b> .....	<b>8</b>
Creating a Dynamic Application .....	9
Creating Collection Objects for a Database Dynamic Application .....	9
Examples of Collection Objects .....	10
<b>Example of a Database Performance Dynamic Application</b> .....	<b>12</b>
Defining the Basic Properties for the Dynamic Application .....	14
Defining the Discovery Object for the Dynamic Application .....	14
Defining the Collection Objects .....	15
Defining the Presentation Objects .....	18
Defining the Threshold Objects .....	20
Defining the Alerts .....	21
Creating a Credential for the MySQL:DBPerformance Dynamic Application .....	25
Aligning the Dynamic Application with a Device .....	26
Viewing Reports for the Dynamic Application .....	27
Viewing Alerts for the Dynamic Application .....	27
Changing the Threshold for a Subscriber Device .....	27
<b>Example of a Dynamic Application of Type "Database Configuration"</b> .....	<b>29</b>
Defining the Basic Properties for the Dynamic Application .....	30
Defining the Discovery Object for the Dynamic Application .....	30
Defining the Collection Objects .....	31
Creating a Credential for the MySQL:DBConfiguration Dynamic Application .....	34
Aligning the Dynamic Application with a Device .....	34
Viewing the Configuration Report for the Dynamic Application .....	35
<b>Example of a Dynamic Application with an Identity-Based Relationship</b> .....	<b>36</b>

Building Dynamic Applications with an Identity-Based Relationship .....	37
Defining the Basic Properties for the Dynamic Application .....	38
Defining the Collection Objects .....	38
Creating the Dynamic Application that Creates the Relationship .....	39
Aligning the Dynamic Applications .....	39
Viewing Dynamic Application Relationships .....	40

---

# Chapter

# 1



## Introduction to Database Dynamic Applications

---

### Overview

This chapter defines Database Dynamic Applications and explains what you can monitor with Database Dynamic Applications.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ()
- To view a page containing all of the menu options, click the Advanced menu icon ()

This chapter covers the following topics:

<i>Prerequisites</i> .....	5
<i>What is a Database Dynamic Application?</i> .....	5
<i>How Do I Allow SL1 to Access the Database?</i> .....	6
<i>What Can I Monitor with a Database Dynamic Application?</i> .....	6
<i>Can I Create My Own Database Dynamic Applications?</i> .....	6
<i>Elements of a Dynamic Application for a Database</i> .....	6

---

## Prerequisites

This manual does not describe elements of Dynamic Application development that are common to all Dynamic Application types. Before reading this manual, you should be familiar with the common elements and concepts of Dynamic Applications. For details on the common elements of Dynamic Applications, see the manual **Dynamic Application Development**.

You should be familiar with the query language used by your database before developing a database Dynamic Application. You must also be familiar with the database schema and the data you want to monitor before developing a database Dynamic Application. For help with these tasks, see your database administrator.

---

## What is a Database Dynamic Application?

Dynamic Applications come in two broad categories, called **archetypes**:

- **Dynamic Applications of archetype Performance.** These Dynamic Applications retrieve trendable (that is, data that can be graphed) performance data from devices or applications. Only this archetype includes the **[Presentations]** tab for defining custom reports. After data has been collected, these reports can be displayed in the Device Management > Performance tab.
- **Dynamic Applications of archetype Configuration.** These Dynamic Applications retrieve configuration data from devices or applications. Data from this archetype can be automatically linked to fields in asset records and can also be displayed in **Hardware Profile** reports and the **Software Found** page. SL1 can automatically monitor one or more data points from this archetype for changes. If the value of the data point changes, SL1 can automatically trigger an event.

**NOTE:** SL1 also includes **Dynamic Applications of archetype Journal**. **The Journal archetype is available only when using the Snippet protocol.** These Dynamic Applications use custom-written Python code to retrieve data from devices or applications. SL1 will display the collected data in log format. Each log entry can contain multiple collected values and can change over time.

These archetypes contain the following types of Dynamic Applications for databases:

- **Database Configuration.** The Dynamic Application retrieves **configuration** data from a database on a managed device. The Dynamic Application includes SQL queries to retrieve data. SL1 executes these queries against a database on each subscriber device. SL1 displays the returned data in configuration tables for each subscriber devices.
- **Database Performance.** The Dynamic Application retrieves **trendable performance** data from a database on a managed device. The Dynamic Application includes SQL queries to retrieve data. SL1 executes these queries against a database on each subscriber device. SL1 displays the returned data in graphs in the **[Performance]** tab for each subscriber device.

SL1 also includes Dynamic Applications for the following protocols: SNMP, SOAP, Snippet (Python), WMI, XML, and XSLT. For an overview of all types of Dynamic Applications see the **Dynamic Application Development**

manual . For details on each protocol, see the manual on that specific protocol (for example, for SNMP, see the *Dynamic Application Development - SNMP* manual).

---

## How Do I Allow SL1 to Access the Database?

For SL1 to successfully send queries to an external database:

- SL1 must have permission to connect to the device that is hosting the database. You might have to perform some configuration tasks on the device or on your firewalls to allow SL1 access.
- If you want SL1 to query an external database, you must configure the appropriate security parameters on the database to give SL1 access to the database.
- SL1 must use a valid database username and database password to query the database. To meet this requirement, SL1 uses **credentials**.

**Credentials** are access profiles (username and password, plus additional information) for external systems. These profiles allow SL1 to access external systems while maintaining the security of the access accounts. Users who need SL1 to retrieve data from these external systems see only the name of the credential, not the username, password, and network information.

For more details on credentials, see the manual *Discovery and Credentials*.

---

## What Can I Monitor with a Database Dynamic Application?

With a Database Dynamic Application, you can monitor any value that can be retrieved with a database query.

However, many Database Dynamic Applications of type Performance query only performance data, like number of reads, number of writes, write time, read time, processor usage, memory usage, number of threads, and buffer cache hits.

Many Database Dynamic Applications of type Configuration query only configuration data, like buffer size, heap size, locks, and number of active users.

---

## Can I Create My Own Database Dynamic Applications?

You can create your own Dynamic Applications to suit your environment and your needs. To create your own Dynamic Applications, you must:

- Determine the data you want to retrieve and monitor.
- Determine the queries you will use to retrieve that data. To write these queries, you must be familiar with the query syntax that is supported by your database, and you must be familiar with the location of data in your database.

---

## Elements of a Dynamic Application for a Database

Database Dynamic Applications have the following elements in common with other Dynamic Application types:

- **Archetypes.** Defines what type of data is being collected and how it will be displayed in SL1. Database Dynamic Applications can be either the *Performance* or *Configuration* archetypes.
- **Properties.** Allows for version control, release notes, collection, and retention settings.
- **Collection Objects.** Define the individual data-points that will be retrieved by the Dynamic Application. These data points are called collection objects. Collection objects define what type of data is being collected (gauge, counter, etc.) and how it is grouped. Collection objects for database Dynamic Applications differ from collection objects in other types of Dynamic Applications. For details on collection objects for database Dynamic Applications, see the section on [Collection Objects](#).
- **Presentations.** For Performance Dynamic Applications, defines how collected values will be displayed by SL1.
- **Thresholds.** Can be used to define a default threshold value that can be included in alerts. The threshold also appears in the **Device Threshold** page for each device the Dynamic Application is aligned with.
- **Alerts.** Alerts allow you to trigger an event based on the values retrieved by the Dynamic Application. If the collected data meets the conditions defined in the alert, the alert can insert a message into device logs and trigger events.
- **Credentials.** Access profiles that define how Dynamic Applications will authenticate on each aligned device and application. Database Dynamic Applications use database credentials. There are multiple ways to align a credential with a Dynamic Application (during discovery, as secondary credentials for a device, or manually in the **Collections** page for a device). For details on how SL1 aligns credentials during discovery and how to manually edit and add new credentials to a device, see the manual **Discovery and Credentials**.
- **Relationships.** Dynamic Applications can be configured to automatically create relationships between devices. For example, the Dynamic Applications in the VMware vSphere and NetApp PowerPacks are configured to create relationships between VMware Datastore component devices and their associated NetApp Volume component devices. Relationships created by Dynamic Applications are used and visualized by the platform in the same manner as relationships created by topology collection, Dynamic Component Mapping, and manually in the user interface. The settings for configuring the creation of relationships in a configuration Database Dynamic Application are the same as the relationship settings for other Dynamic Application protocols.

---

# Chapter

# 2

## Defining Collection Objects

---

### Overview



A **collection object** is an individual data-point that will be collected by a Dynamic Application. Most Dynamic Applications collect multiple data-points. These data-points are referred to as **objects**.

For Database Dynamic Applications, each collection object is populated with the results of a query.

For example, suppose you want to monitor the write-speed of the database every 15 minutes. You could define your Dynamic Application to execute every 15 minutes. You could define a collection object called "write\_speed". You could then define a query that retrieves the value for "write speed" (usually stored in one of the administrative tables). You could align this query with the "write\_speed" object and populate the object.

**NOTE:** This chapter describes only the fields specific to collection objects for a database Dynamic Application. All the remaining fields, for both performance and configuration archetypes, are described in detail in the *Dynamic Application Development* manual .

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ().
- To view a page containing all of the menu options, click the Advanced menu icon (  ).

This chapter covers the following topics:

<a href="#">Creating a Dynamic Application</a> .....	9
<a href="#">Creating Collection Objects for a Database Dynamic Application</a> .....	9
<a href="#">Examples of Collection Objects</a> .....	10



---

## Creating a Dynamic Application

To create a Dynamic Application, you must:

1. Define the general properties of the Dynamic Application.
2. **Define the collection objects you want to monitor.**
3. Optionally, define thresholds for the values of collection objects.
4. Optionally, define graphs of the values of collection objects.
5. Optionally, define alerts that are triggered by the values of collection objects.


All these steps, except for defining collection objects, are the same for all types of Dynamic Applications. The steps that are the same for all types of Dynamic Applications are described in the manual **Dynamic Application Development**.

The step that is specific to database Dynamic Applications, defining collection objects, is described in this section.

---

## Creating Collection Objects for a Database Dynamic Application

This section describes how to define a collection object for a Dynamic Application of type Database Performance or Database Configuration. This section describes only the fields specific to a database Dynamic Application. All the remaining fields, for both performance and configuration archetypes, are described in detail in the manual **Dynamic Application Development**.

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. **If the Dynamic Application already exists**, in the **Dynamic Applications Manager** page, find the Dynamic Application for which you want to define a new object. Select its wrench icon ()
3. **To create a new Dynamic Application**, follow the steps in the manual **Dynamic Application Development**. When defining a database Dynamic Application, most of the steps are the same for all types of Dynamic Applications. Only one step, creating collection objects, is unique to database Dynamic Applications. That unique step is described in this section.
4. Select the **[Collections]** tab.
5. In the **Collection Objects** page, enter a value in each field in the top pane. This section describes only the fields specific to a database Dynamic Application. All the remaining fields are described in detail in the manual **Dynamic Application Development**.

- **SQL Query.** Enter a valid database query in this field. The value retrieved by this query will be stored in the collection object. Be sure the query has been tested on the intended database before you include it in a Dynamic Application. Also, be sure the query can be resolved within the polling frequency of the Dynamic Application.

If a single query will return multiple columns, and you want to view the output from multiple columns and view graphs for multiple columns, you must **create a collection object for each returned column**. This means that for each collection object, you must enter the same query and define a different **Object Name**, **Class Type**, and a different **Trended Column** for each returned column.

- **Trended Column.** If a query returns multiple columns, enter the name of the column that you want to see graphed in Device Management > Performance. If a query returns multiple columns and this field is left blank for each collection object, by default only the first returned column will be graphed in the Device Management > Performance page.

6. Select the **[Save]** button to save the new collection object.

7. Repeat these steps for each collection object you want to define for the Dynamic Application.

---

## Examples of Collection Objects

- For **MySQL**, we could enter one of the following queries into the **SQL Query** field:

- To retrieve the version number of a MySQL database:

```
show global variables like 'version'
```

- To retrieve the number of active events on the SL1 system:

```
SELECT count( * ) Value FROM master_events.events_active;
```

- For **MS SQL**, we could enter one of the following queries into the **SQL Query** field:

- To retrieve the name of the MS SQL database:

```
select name from sys.databases
```

- To retrieve the number of logins to the database:

```
select cntr_value from sysperfinfos where counter_name='Logins/sec'
```

- For **Oracle**, we could enter one of the following queries into the **SQL Query** field:

- To retrieve the number of times a process was delayed while waiting to access the rollback segment

```
SELECT (Sum(waits) / Sum(gets)) * 100 FROM v$rollstat
```

- To retrieve the hit ratio of requests to the block buffer (versus "hits" to the physical disk):

```
SELECT (1 - (phys.value / (db.value + cons.value))) * 100 FROM  
v$sysstat phys,v$sysstat db,v$sysstat cons WHERE phys.name =  
'physical reads' AND db.name = 'db block gets' AND cons.name =  
'consistent gets'
```

---

# Example

# 1

## Example of a Database Performance Dynamic Application

---

### Overview

In this chapter, we will walk through a Dynamic Application that monitors the performance of a MySQL database.



The **MySQL DB Performance** Dynamic Application makes multiple queries to an internal table in MySQL. This internal table stores status information about the MySQL server, such as number of connections, information about index scans, information about pages in the buffer pool, reads and writes to key blocks, number of open files, information on the query cache, number of slow queries, information about table locks, and information about threads.

The **MySQL DB Performance** Dynamic Application includes presentation objects, threshold objects, and alert definitions that allow you to monitor the status of a MySQL server.

In this chapter, we have aligned the Dynamic Application with the Database Server. The Database Server uses a MySQL database.

**NOTE:** The **MySQL DB Performance** Dynamic Application includes multiple Collection Objects, Presentation Objects, Threshold Objects, and Alerts. This chapter will walk you through only two of each type of object.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon (.
- To view a page containing all of the menu options, click the Advanced menu icon (  ).

This chapter covers the following topics:

[Defining the Basic Properties for the Dynamic Application](#) .....14

<i>Defining the Discovery Object for the Dynamic Application</i> .....	14
<i>Defining the Collection Objects</i> .....	15
<i>Defining the Presentation Objects</i> .....	18
<i>Defining the Threshold Objects</i> .....	20
<i>Defining the Alerts</i> .....	21
<i>Creating a Credential for the MySQL:DBPerformance Dynamic Application</i> .....	25
<i>Aligning the Dynamic Application with a Device</i> .....	26
<i>Viewing Reports for the Dynamic Application</i> .....	27
<i>Viewing Alerts for the Dynamic Application</i> .....	27
<i>Changing the Threshold for a Subscriber Device</i> .....	27

---

## Defining the Basic Properties for the Dynamic Application

To create the container for this Dynamic Application and define its general properties, perform the following:

**NOTE:** For details on each field and its possible options, see the manual *Dynamic Application Development*.

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Click the [Actions] button, and then select **Create New Dynamic Application**. The **Dynamic Applications Create New Application** page appears.
3. Supply values in the following fields:
  - **Application Name**. Enter *MySQL:DBPerformance*.
  - **Application Type**. The protocol SL1 will use and the type of data that will be collected. Select *Database Performance*.
  - **Version Number**. Accept the default value. You can customize this value and increment it according to your change-management policies.
  - **Operational State**. Specifies whether SL1 will collect data from devices using this Dynamic Application. This field also specifies whether SL1 will automatically align this Dynamic Application to devices during discovery, re-discovery, and nightly auto-discovery. Select *Enabled*.
  - **Poll Frequency**. Frequency at which SL1 will poll devices that use this Dynamic Application. Select *"Every 5 Minutes"*, so we can quickly view retrieved data in this example.
  - **Abandon Collection**. Accept the default value. Specifies how many collection objects must be unavailable before the Dynamic Application should stop trying to collect data and wait until the next scheduled collection session. *Default* specifies a threshold of two collection objects.

**NOTE:** For all objects except those retrieved from a database, the timeout limit is specified in the credential. For database objects, the timeout limit is specified internally by SL1.

- **Context**. Leave this field blank.
4. Click the [Save] button to save the Dynamic Application.

---


## Defining the Discovery Object for the Dynamic Application

A **discovery object** is a type of collection object. If you want SL1 to automatically align devices with a Dynamic Application during discovery, you must include a discovery object in that Dynamic Application.

**NOTE:** For more details on discovery objects, see the manual *Dynamic Application Development*.

To create a discovery object for the Dynamic Application **MySQL:DBPerformance**, we will write a query that will return a value only if MySQL is running on a device.

To create the discovery object:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Click the wrench icon (  ) for the Dynamic Application named **MySQL:DBPerformance**.
3. Click the **[Collections]** tab. The **Dynamic Applications | Collections Objects** page (System > Manage > Applications > Collections) appears.
4. Supply values in the following fields:
  - **Object Name.** Enter *Discovery Object*.
  - **SQL Query.** This field specifies the query that SL1 will use to collect the discovery object. We will query a value that indicates that a MySQL database exists, regardless of the status of the data in the database. Enter the following query:

```
show global status like 'Connections'
```

- This query searches an internal table that stores information about all connections to the MySQL database (`show global status`).
  - The query retrieves the value of the status variable "Connections" (`like 'Connections'`).
  - The variable "Connections" contains the number of connection attempts (successful or not) to the MySQL server. The value can be zero ("0") or greater.
  - This query will return a value even if no connections have been made previously.
- **Class Type.** Select *[100] Discovery*.
5. Click the **[Save]** button to save the collection object. Because this collection object has been defined with a **Class Type** of *[100] Discovery*, the user interface displays additional fields that are specific to discovery objects.
  6. Enter values in the following fields:
    - **Alignment Condition.** Specifies how this discovery object should be evaluated. Select *Align if OID is present*. This choice tells SL1 to automatically align the Dynamic Application with each device that returns a value for the query in the **SQL Query** field.
    - **Validity Check.** Leave blank.

---

## Defining the Collection Objects

The **MySQL:DBPerformance** Dynamic Application on [the ScienceLogic Support Site](#) includes 27 Collection Objects. This section will walk you through the creation of only two collection objects.

**NOTE:** For more details on collection objects, see the manual *Dynamic Application Development*.

In this section, we will create the following collection objects:

- **Key\_reads**. This collection object monitors the MySQL status variable `key_reads`. This status variable specifies the number of times MySQL had to access the file system (instead of the key cache) to fetch database indexes. If `key_reads` is large, then the key buffer is probably too small.
- **Key\_read\_requests**. This collection object monitors the MySQL status variable `key_read_requests`. This status variable specifies the total number of requests to read a key block from the cache.

If MySQL must fetch database indexes from the filesystem, queries to that database will be slower than usual. If your MySQL server must frequently fetch database indexes from the filesystem, you should increase the size of the key buffer.

To create these two collection objects, perform the following:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Click the wrench icon (🔧) for the Dynamic Application named **MySQL:DBPerformance**.
3. Click the **[Collections]** tab. The **Dynamic Applications | Collections Objects** page (System > Manage > Applications > Collections) appears.
4. First, we will define the **Key\_reads** collection object. This collection object monitors the MySQL status variable `key_reads`. This status variable specifies the number of times MySQL had to access the file system (instead of the key cache) to fetch database indexes.
5. Supply values in the following fields:
  - **Object Name**. Enter `Key_reads`.
  - **SQL Query**. Enter the following:

```
show global status like 'Key_reads'
```

- This query searches an internal table that stores information about the MySQL server (`show global status`).
  - The query retrieves the value of the status variable "Key reads" (`like 'Key_reads'`).
  - The variable `Key_reads` contains the number of times MySQL had to access the file system (instead of the key cache) to fetch database indexes.
- **Class Type**. Select *1 Performance Counter*.
  - **Group Number**. Select *No Group*, and leave the second drop-down as *Standard*.
  - **Trended Column**. Enter *Value*. The query returns two columns: *Variable\_name*, which contains the name of the variable (`Key_reads`) and *Value*, which contains the value of the variable. We are interested only in the value in the *Value* column. We want SL1 to graph the value from the *Value* column.
  - **Enable Deviation Alerting**. Do not select these checkboxes.



- **Description.** Leave blank.
  - **Formula.** Leave blank.
6. Click the **[Save]** button to save the new collection object.
  7. Next, we will define the **Key\_read\_requests** collection objects. This collection object monitors the MySQL status variable `key_read_requests`. This status variable specifies the total number of requests to read a key block from the cache.
  8. Supply values in the following fields:
    - **Object Name.** Enter `Key_read_requests`.
    - **SQL Query.** Enter the following:
 

```
show global status like 'Key_read_requests'
```

      - This query searches an internal table that stores information about the MySQL server (`show global status`).
      - The query retrieves the value of the status variable "Key\_read\_requests" (`like 'Key_read_requests'`).
      - The variable `Key_read_requests` contains the total number of requests to read a key block from the cache.
    - **Class Type.** Select *1 Performance Counter*.
    - **Group Number.** Select *No Group*, and leave the second drop-down as *Standard*.
    - **Trended Column.** Enter *Value*. The query returns two columns: *Variable\_name*, which contains the name of the variable (`Key_read_requests`) and *Value*, which contains the value of the variable. We are interested only in the value in the *Value* column. We want SL1 to graph the value from the *Value* column.
    - **Enable Deviation Alerting.** Do not select these checkboxes.
    - **Description.** Leave blank.
    - **Formula.** Leave blank.
  9. Click the **[Save]** button to save the new collection object.
  10. In our example, you will notice that the collection objects have the following object IDs:
    - `Key_reads = o_5500`
    - `Key_read_requests = o_5501`

**NOTE:** *On your SL1 system, the collection objects will have different object IDs.* Whether you have imported the Dynamic Application or are creating the Dynamic Application from the steps in this chapter, the collection objects will have different object IDs than on our example SL1 system.

---

## Defining the Presentation Objects

Presentation objects allow you to define how SL1 should use the values collected by the Dynamic Application to create performance graphs.


**NOTE:** For more details on presentation objects, see the manual *Dynamic Application Development*.

The **MySQL:DBPerformance** Dynamic Application on [the ScienceLogic Support Site](#) includes 33 Presentation Objects. This section will walk you through the creation of only two presentation objects.

In this section, we will create the following presentation objects:

- **Key\_reads**. Displays the value of the **Key\_reads** collection object, over time. The **Key\_reads** collection object specifies the number of times MySQL had to access the file system (instead of the key cache) to fetch database indexes. If the value of the **Key\_reads** collection object is large, then the key buffer is probably too small. The **Key\_reads** presentation object will graph each collected value of the **Key\_reads** collection object and its associated date and time.
- **Key\_read\_requests**. Displays the value of the **Key\_read\_requests** collection object, over time. The **Key\_read\_requests** collection object specifies the total number of requests to read a key block from the cache. The **Key\_read\_requests** presentation object will graph each collected value of the **Key\_read\_requests** collection object and its associated date and time.

To create these two presentation objects, perform the following:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Click the wrench icon (  ) for the Dynamic Application named **MySQL:DBPerformance**.
3. Click the **[Presentations]** tab. The **Dynamic Applications Presentation Objects** page appears.
4. First, we will define the **Key\_reads** presentation object. The **Key\_reads** presentation object will graph each collected value of the **Key\_reads** collection object and its associated date and time.
5. Supply values in the following fields:
  - **Report Name**. Enter *Key\_reads*. This name will appear in the NavBar of the **Performance** page for each device that subscribes to the Dynamic Application. This name will also appear as a title for the graph.
  - **Active State**. Select *Enabled*. SL1 will immediately create the graph at the next polling session.
  - **Data Unit**. Leave blank.
  - **Abbreviation/Suffix**. Leave blank.
  - **Show as Percent**. Select *No*.
  - **Vitals Link**. Select *Disabled*.
  - **Formula Editor**. Enter the following: `(o_5500)`
    - This formula tells SL1 to graph each value of the collection object **o\_5500**. In our example system, this is the object ID for the **Key\_reads** collection object. SL1 will graph each value of this collection object, along with its associated date and time.

**NOTE:** The object ID for the **Key\_reads** collection object will be different on your SL1 system. If you are creating a new Dynamic Application using the instructions in this chapter, please enter the object ID for the **Key\_reads** collection object, as it appears on your SL1 system.

6. Click the **[Save]** button to save the new presentation object.
7. Next, we will define the **Key\_read\_requests** presentation object. The **Key\_read\_requests** presentation object will graph each collected value of the **Key\_read\_requests** collection object and its associated date and time.
8. Supply values in the following fields:
  - **Report Name.** Enter *Key\_read\_requests*. This name will appear in the NavBar of the **Performance** page for each device that subscribes to the Dynamic Application. This name will also appear as a title for the graph.
  - **Active State.** Select *Enabled*. SL1 will immediately create the graph at the next polling session.
  - **Data Unit.** Leave blank.
  - **Abbreviation/Suffix.** Leave blank.
  - **Show as Percent.** Select *No*.
  - **Vitals Link.** Select *Disabled*.
  - **Formula Editor.** Enter the following: (o\_5501)
    - This formula tells SL1 to graph each value of the collection object **o\_5501**. On our example SL1 system, this is the object ID for the **Key\_read\_requests** collection object. SL1 will graph each value of this collection object, along with its associated date and time.

**NOTE:** The object ID for the **Key\_read\_requests** collection object will be different on your SL1 system. If you are creating a new Dynamic Application using the instructions in this chapter, please enter the object ID for the **Key\_read\_requests** collection object, as it appears on your SL1 system.

9. Click the **[Save]** button to save the new presentation object.

---

## Defining the Threshold Objects

A threshold object is an object that you can use in the formula for an alert definition or a presentation object, just as you would use a collection object.


Threshold objects can also appear as thresholds in the **Device Thresholds** page (Devices > Classic Devices > wrench icon > Thresholds, or Registry > Devices > Device Manager > wrench icon > Thresholds in the classic SL1 user interface) for each device that subscribes to the Dynamic Application.

**NOTE:** For more details on threshold objects, see the manual *Dynamic Application Development*.

The **MySQL:DBPerformance** Dynamic Application on [the ScienceLogic Support Site](#) includes seven threshold objects. This section will walk you through the creation of only one threshold object.

- **Keycache\_hitrate**. We will use this threshold in the formula for two alerts. The initial alert is triggered when MySQL fetches database indexes from the filesystem instead of from the key cache. This threshold defines the percentage of fetches that can be from the filesystem instead of from the key cache before SL1 generates an alert.

To create this threshold object, perform the following:

1. Go to the **System > Manage > Applications** page (System > Manage > Applications).
2. Click the wrench icon (  ) for the Dynamic Application named **MySQL:DBPerformance**
3. Click the **[Thresholds]** tab. The **Dynamic Applications Threshold Objects** page appears.
4. Supply values in the following fields:
  - **Threshold Name**. Enter *Keycache\_hitrate*.
  - **Override Threshold Value**. Select *Enabled*. This threshold will appear in the the **Device Thresholds** page (Devices > Classic Devices > wrench icon > Thresholds, or Registry > Devices > Device Manager > wrench icon > Thresholds in the classic SL1 user interface for devices that subscribe to the Dynamic Application.
  - **Numeric Range: High**. Enter *100*. By default, the highest possible value for this threshold will be "100". This value will appear at the high end of the slider in the the **Device Thresholds** page (Devices > Classic Devices > wrench icon > Thresholds, or Registry > Devices > Device Manager > wrench icon > Thresholds in the classic SL1 user interface.
  - **Numeric Range: Low**. Enter *0*. By default, the lowest possible value for this threshold will be "0". This value will appear at the low end of the slider in the the **Device Thresholds** page (Devices > Classic Devices > wrench icon > Thresholds, or Registry > Devices > Device Manager > wrench icon > Thresholds in the classic SL1 user interface.
  - **Threshold Type**. Select *Percentage*.
  - **Threshold Value**. Enter *99*.
5. Click the **[Save]** button to save the new threshold.
6. If you imported the Dynamic Application from [the ScienceLogic Support Site](#), you will notice that the threshold object has the following object ID:
  - **Keycache\_hitrate = t\_130**

**NOTE:** *On your SL1 system, the threshold object will have different object IDs.* Whether you have imported the Dynamic Application or are creating the Dynamic Application from the steps in this chapter, the collection objects will have different object IDs than on our example SL1 system.

---

## Defining the Alerts

Alerts allow you to examine and manipulate values retrieved by a Dynamic Application. An alert defines the conditions during which you would like SL1 to insert a message in the device log. You can define events that are triggered when the alert message appears in a device log.

**NOTE:** For details on alerts, see the manual *Dynamic Application Development*.

The **MySQL:DBPerformance** Dynamic Application on [the ScienceLogic Support Site](#) includes 14 alerts. This section will walk you through the creation of only two alerts.

In this section, we will create the following alerts:

- **MySQL:Keycache\_hitrate\_low**. This alert compares the value of the **Key\_reads** collection object to the value of the **Key\_read\_requests** collection object. Remember that the **Key\_reads** collection object specifies the number of times the MySQL server has had to fetch a database index from the file system instead of from the cache. The **Key\_read\_requests** object specifies the number of times the MySQL server has fetched a database index from cache. The alert says "If the number of times MySQL has had to fetch a database index from the file system is 1% or more of the number of times MySQL has fetched a database index from cache, generate an alert of severity "minor".
- **MySQL:Keycache\_hitrate\_normal**. This alert compares the value of the **Key\_reads** collection object to the value of the **Key\_read\_requests** collection object. The alert says "If the **MySQL:Keycache\_hitrate\_low** alert is still active, and if the number of times MySQL has had to fetch a database index from the file system is less than 1% of the number of times MySQL has fetched a database index from cache, generate an alert of severity "healthy".

To create these two alerts, perform the following:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Click the wrench icon (🔧) for the Dynamic Application named **MySQL:DBPerformance**.
3. Click the **[Alerts]** tab. The **Dynamic Applications Alert Objects** page appears.
4. First, we will define the **MySQL:Keycache\_hitrate\_low** alert.
5. Supply values in the following fields:
  - **Policy Name**. Enter *MySQL:Keycache\_hitrate\_low*.
  - **Active State**. Select *Enabled*. SL1 will monitor this alert.
  - **Log Message**. If this alert evaluates to TRUE, the alert will insert the following message in the device log (on the device where the condition occurred). Enter the following:

```
MySQL Keycache hitrate: %V% is low. Threshold: %T.
```

    - The **%V** variable says "substitute the value returned by the **result** function".
    - The **%T** variable says "substitute the value returned by the **threshold** function".
  - **Maintain State**. Select *Yes*. This alert will maintain its state until it is explicitly cleared by an event.
  - **Trigger Alert**. Select *None*. This is a deprecated field.
  - **Formula Editor**. This is where you describe the conditions under which you want SL1 to make an entry in the device log. Enter the following:

```
result(100-((o_5500/o_5501)*100)) < threshold(t_191)
```

- This formula says: Divide the number of **Key\_reads** by the number of **Key\_read\_requests** and convert that value to percent. If the percentage of **Key\_Reads** is 1% or more, the alert will evaluate to TRUE. When the alert evaluates to TRUE, it makes an entry of severity "minor" in the appropriate device log.
- **o\_5500** is the object ID of the **Key\_reads** collection object.

**NOTE:** On your SL1 system, the **Key\_reads** collection object will have a different object ID. Substitute the object ID from your SL1 system.

- **o\_5501** is the object ID of the **Key\_read\_requests** collection object. If you have created the **MySQL:DBPerformance** Dynamic Application manually (instead of importing the Dynamic Application from [the ScienceLogic Support Site](#)), the **Key\_read\_requests** collection object will have a different object ID on your SL1 system. Substitute the object ID from your SL1 system.

**NOTE:** On your SL1 system, the **Keycache\_hitrate** threshold will have a different object ID. Substitute the object ID from your SL1 system.

- **t\_191** is the object ID of the **Keycache\_hitrate** threshold.

**NOTE:** On your SL1 system, the **Keycache\_hitrate** threshold will have a different object ID. Substitute the object ID from your SL1 system.

- Remember that we set the **Keycache\_hitrate** threshold to "99".
- The **result** function returns the value of the formula and stores the value of the formula in the **%V** variable.
- The **threshold** function returns the value of the threshold variable and stores the value of the threshold variable in the **%T** variable.

6. Click the **[Save]** button to save the alert.

7. In our example SL1 system, you will notice that the alert object has the following object ID:

- **MySQL:Keycache\_hitrate\_low = a\_920**

**NOTE:** On your SL1 system, the alert object will have a different object ID.

8. Next we will define the **MySQL:Keycache\_hitrate\_normal** alert.

9. Supply values in the following fields:

- **Policy Name.** Enter *MySQL:Keycache\_hitrate\_normal*.
- **Active State.** Select *Enabled*. SL1 will monitor this alert.

- **Log Message.** If this alert evaluates to TRUE, the alert will insert the following message in the device log (on the device where the condition occurred. )Enter the following:

```
MySQL Keycache hitrate: %V% is normal.
```

- The **%V** variable says "substitute the value returned by the **result** function".
- **Maintain State.** Select *No*. This alert will not maintain its state and does not need to be explicitly cleared by an event.
- **Trigger Alert.** Select *None*. This is a deprecated field.
- **Formula Editor.** This is where you describe the conditions under which you want SL1 to make an entry in the device log. Enter the following:

```
result(100-((o_5500/o_5501)*100)) >= threshold(t_191) and active  
(a_920)
```

- This formula says: Divide the number of **Key\_reads** by the number of **Key\_read\_requests** and convert that value to percent. If the alert **MySQL:keycache\_hitrate\_low** is still active, and if the percentage of **Key\_Reads** is 1% or less, the alert will evaluate to TRUE. When the alert evaluates to TRUE, it makes an entry of severity "healthy" in the appropriate device log.
- **o\_5500** is the object ID of the **Key\_reads** collection object.

**NOTE:** On your SL1 system, the **Key\_reads** collection object will have a different object ID. Substitute the object ID from your SL1 system.

- **o\_5501** is the object ID of the **Key\_read\_requests** collection object.

**NOTE:** On your SL1 system, the **Key\_read\_requests** collection object will have a different object ID. Substitute the object ID from your SL1 system.

- **t\_191** is the object ID of the **Keycache\_hitrate** threshold.

**NOTE:** On your SL1 system, the **Keycache\_hitrate** threshold will have a different object ID . Substitute the object ID from your SL1 system.

- **a\_920** is the object ID of the alert **MySQL:keycache\_hitrate\_low**.

**NOTE:** On your SL1 system, the **Keycache\_hitrate\_low** alert will have a different object ID. Substitute the object ID from your SL1 system.



- Remember that we set the **Keycache\_hitrate** threshold to "99".
  - The **result** function returns the value of the formula and stores the value of the formula in the **%V** variable.
  - The **threshold** function returns the value of the threshold variable and stores the value of the threshold variable in the **%T** variable.
  - The **active** function checks the state of a specified alert. If the specified alert is still active, the **active** function returns the value TRUE.
10. Click the **[Save]** button to save the alert.
  11. On our example SL1 system, notice that the alert object has the following object ID:
    - **MySQL:Keycache\_hitrate\_normal = a\_921**

**NOTE:** On your SL1 system, the alert object will have a different object ID.

---

## Creating a Credential for the MySQL:DBPerformance Dynamic Application

If you want to align the **MySQL:DBPerformance** Dynamic Application with SL1's MySQL database, you must create a database credential that allows access to SL1's MySQL database.

Before you define the credential, you must collect the information you will need. In this case, you will need the username and password for the MySQL database. This account was defined during setup and is different than the account for logging into SL1.

If you can log in to the phpMyAdmin tool from the **Appliance Manager** page (System > Settings > Appliances), you can use the username and password you used to access the phpMyAdmin tool as the username and password in the credential. For details on accessing the phpMyAdmin tool, see the manual **System Administration**. If you need help, ask your administrator.

For details on the database password, see the manual **System Administration**.

For details on credentials, see the manual **Discovery and Credentials**.

To create the credential for this example:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. In the the **Credential Management** page, click the **[Create]** menu. Select *Database Credential*.
3. The **Credential Editor** modal page appears. In this page, you can define the new database credential. To define the new credential, supply values in the following fields:
  - **Profile Name**. Enter *EM7 DB*.
  - **DB Type**. Select *MySQL*.
  - **DB Name**. Enter *master*.
  - **DB User**. Username associated with a valid account on the database.
  - **Password**. Password associated with a valid account on the database.

- **Hostname/IP.** Hostname or IP address where the database resides. To use the localhost, in the **Hostname/IP** field, enter the IP address `127.0.0.1`. The credential will not work if you enter the string `localhost` in the **Hostname/IP** field.
- **Port.** Enter `7706`.

4. Click the **[Save]** button to save the new database credential.

---

## Aligning the Dynamic Application with a Device


For our example, we aligned the **MySQL:DBPerformance** Dynamic Application with the Database Server.

There are three ways to align the **MySQL:DBPerformance** Dynamic Application with a device:

- **During initial discovery or nightly auto-discovery.** Because the **MySQL:DBPerformance** Dynamic Application includes a discovery object, SL1 can automatically align this Dynamic Application with devices during discovery. For details on discovery, see the manual **Discovery and Credentials**.
- **Manual discovery for the Dynamic Application.** Because the **MySQL:DBPerformance** Dynamic Application includes a discovery object, SL1 can automatically align this Dynamic Application with devices. From the **Dynamic Applications Manager** page, you can manually execute discovery for all devices, but only for the **MySQL:DBPerformance** Dynamic Application. For details on how to perform this type of discovery, see the manual **Dynamic Application Development**.
- **Manually associate the Dynamic Application with an existing device.** For details on how to perform this type of discovery, see the manual **Dynamic Application Development**.

This section will walk you through the steps to manually align the **MySQL:DBPerformance** Dynamic Application with the Database Server or the All-In-One Appliance.

To manually align the Dynamic Application to the Database Server or the All-In-One Appliance:


1. Go to the **Appliance Manager** page (System > Settings > Appliances). Determine the device name of the Database Server or the All-In-One Appliance.
2. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
3. Find the device where the database resides (either the Database Server or the All-In-One Appliance). Click the device's wrench icon (.
4. The **Device Properties** page appears. Click the **[Collections]** tab.
5. In the **Dynamic Application Collections** page, click the **[Action]** menu and select *Add Dynamic Application*.
6. The **Dynamic Application Alignment** page appears. Select **MySQL:DBPerformance** in the **Dynamic Applications** pane, and select **EM7 DB** in the **Credentials** pane.
7. Click the **[Save]** button to add the Dynamic Application to the device.
8. At the next polling period, SL1 should start collecting the data specified in the **MySQL:DBPerformance** Dynamic Application from the device where SL1's database resides.

---

## Viewing Reports for the Dynamic Application

For our example, we aligned the **MySQL:DBPerformance** Dynamic Application with the Database Server on our device em7\_ao.


To view the graphs for the presentation objects **key\_reads** and **key\_read\_requests**:

1. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
2. Find the device for which you want to edit thresholds. Click its bar-graph icon (.
3. Click the **[Performance]** tab. The **Device Performance** page appears.
4. In the NavBar, find the entry for the **MySQL:DBPerformance** Dynamic Application and expand it. Select the entry for **Key\_reads**.
5. The graph for **Key\_reads** appears.
6. The graph displays the value of the **Key\_reads** collection object in the y-axis and the date and time in the x-axis.
7. In the NavBar, find the entry for the **MySQL:DBPerformance** Dynamic Application and expand it. Select the entry for **Key\_read\_requests**.
8. The graph for **Key\_read\_requests** appears.
9. The graph displays the value of the **Key\_read\_requests** collection object in the y-axis and the date and time in the x-axis.

---

## Viewing Alerts for the Dynamic Application

To view alerts for the Dynamic Application:


1. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
2. Find the subscriber device for which you want to edit thresholds. Click its wrench icon (.
3. Click the **[Logs]** tab. The **Device Logs & Messages** page appears.
4. Look for alert messages from the Dynamic Application. In our example, our MySQL database did not fetch any database indexes from the file system, so only the "normal" alert appears in our device log.

---

## Changing the Threshold for a Subscriber Device

You can change one or more threshold values for a single device. When SL1 evaluates alerts **for that device**, it will use the threshold values set in the **Device Threshold** page instead of the threshold value set in the **Dynamic Application Threshold Objects** page.

To edit a threshold for a single device:

1. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
2. Find the device for which you want to edit thresholds. Click its wrench icon (.
3. Click the **[Thresholds]** tab. The **Device Thresholds** page appears.
4. In **Device Thresholds** page, move the sliders to edit one or more thresholds.
5. To save your changes, click the **[Save]** button.

---

# Example

# 2

## Example of a Dynamic Application of Type "Database Configuration"

---



### Overview

In this chapter, we will walk through a Dynamic Application that monitors the performance of a MySQL database.

This chapter will walk you through the Dynamic Application *MySQL:DBConfiguration*. The *MySQL:DBConfiguration* Dynamic Application makes multiple queries to an internal table in MySQL. This internal table stores variables that contain configuration information for the MySQL server, such as MySQL version, the configuration of the auto-commit feature, the maximum allowed number of connections, and the sizes of various caches.

In this chapter, we have aligned the Dynamic Application with the Database Server. The Database Server uses a MySQL database.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon (.
- To view a page containing all of the menu options, click the Advanced menu icon (  ).

This chapter covers the following topics:

<i>Defining the Basic Properties for the Dynamic Application</i> .....	30
<i>Defining the Discovery Object for the Dynamic Application</i> .....	30
<i>Defining the Collection Objects</i> .....	31
<i>Creating a Credential for the MySQL:DBConfiguration Dynamic Application</i> .....	34
<i>Aligning the Dynamic Application with a Device</i> .....	34
<i>Viewing the Configuration Report for the Dynamic Application</i> .....	35

---

## Defining the Basic Properties for the Dynamic Application

To create the container for this Dynamic Application and define its general properties, perform the following:

**NOTE:** For details on each field and its possible options, see the manual *Dynamic Application Development*.

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Select the **[Actions]** button, and then select *Create New Dynamic Application*. The **Dynamic Applications Create New Application** page appears.
3. Supply values in the following fields:
  - **Application Name.** Enter *MySQL:DBConfiguration*.
  - **Application Type.** The protocol SL1 will use and the type of data that will be collected. Select *Database Configuration*.
  - **Version Number.** Accept the default value. You can customize this value and increment it according to your change-management policies.
  - **Operational State.** Specifies whether SL1 will collect data from devices using this Dynamic Application. This field also specifies whether SL1 will automatically align this Dynamic Application to devices during discovery, re-discovery, and nightly auto-discovery. Select *Enabled*.
  - **Poll Frequency.** Frequency at which SL1 will poll devices that use this Dynamic Application. Select *"Every 2 Minutes"*, so we can quickly view retrieved data in this example.
  - **Abandon Collection.** Accept the default value. Specifies how many collection objects must be unavailable before the Dynamic Application should stop trying to collect data and wait until the next scheduled collection session. *Default* specifies a threshold of two collection objects.

**NOTE:** For all objects except those retrieved from a database, the timeout limit is specified in the credential. For database objects, the timeout limit is specified internally by SL1.

- **Context.** Leave this field blank.
4. Select the **[Save]** button to save the Dynamic Application.

---

## Defining the Discovery Object for the Dynamic Application

A **discovery object** is a type of collection object. If you want SL1 to automatically align devices with a Dynamic Application during discovery, you must include a discovery object in that Dynamic Application.

**NOTE:** For more details on discovery objects, see the manual *Dynamic Application Development*.

To create a discovery object for the Dynamic Application **MySQL: DBConfiguration**, we will write a query that will return a value only if MySQL is running on a device.

To create the discovery object:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Select the wrench icon (🔧) for the Dynamic Application named **MySQL:DB Configuration**.
3. Select the **[Collections]** tab. The **Dynamic Applications | Collections Objects** page appears.
4. Supply values in the following fields:
  - **Object Name.** Enter *Discovery Object*.
  - **SQL Query.** This field specifies the query that SL1 will use to collect the discovery object. We will query a value that indicates that a MySQL database exists, regardless of the status of the data in the database. Enter the following query:

```
show global variables like 'version'
```

- This query searches an internal table that stores configuration information about the MySQL database (`show global variables`).
  - The query retrieves the value of the status variable "version" (`like 'version'`).
  - The variable "version" contains the version number and name of the MySQL server.
  - This query will return a value only if MySQL is installed on a device. This query is not dependent on the data stored in the database.
- **Class Type.** Select *[100] Discovery*.
5. Select the **[Save]** button to save the collection object and the page will change to appear as it does below. Because this collection object has been defined with a **Class Type** of *[100] Discovery*, SL1 displays additional fields that are specific to discovery objects.
  6. Enter values in the following fields:
    - **Alignment Condition.** Specifies how this discovery object should be evaluated. Select *Align if OID is present*. This choice tells SL1 to automatically align the Dynamic Application with each device that returns a value for the query in the **SQL Query** field.
    - **Validity Check.** Leave blank.

---

## Defining the Collection Objects

This section will walk you through the creation of only six collection objects for the **MySQL: DBConfiguration** Dynamic Application.

**NOTE:** For more details on collection objects, see the manual *Dynamic Application Development*.

In this section, we will create the following collection objects:

- **mysql\_version.** This collection object monitors the MySQL configuration variable *version*. This configuration variable contains the version number and version name of the MySQL software.
- **auto\_commit.** This collection object monitors the MySQL configuration variable *autocommit*. This configuration variable specifies whether the autocommit mode is enabled (set to 1). If autocommit is enabled, users do not have to commit each transaction. If autocommit is not enabled, users must either commit each transaction to save it to the database.
- **max\_connections.** This collection object monitors the MySQL configuration variable *max\_connections*. This configuration variable specifies the maximum allowed number of client connections to the database.
- **key\_cache\_size.** This collection object monitors the MySQL configuration variable *key\_buffer\_size*. This configuration variable specifies the size of the cache where index information is stored.
- **query\_cache\_size.** This collection object monitors the MySQL configuration variable *query\_cache\_size*. This configuration variable specifies the size of the cache where query results are stored.
- **thead\_cache\_size.** This collection object monitors the MySQL configuration variable *thread\_cache\_size*. This configuration variable specifies the number of threads that should be saved in the thread cache.

To create these collection objects, perform the following:

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Select the wrench icon (🔧) for the Dynamic Application named **MySQL: DBConfiguration**.
3. Select the **[Collections]** tab. The **Dynamic Applications | Collections Objects** page appears.
4. First, we will define the **mysql\_version** collection object. This collection object monitors the MySQL configuration variable *version*. This configuration variable contains the version number and version name of the MySQL software.
5. Supply values in the following fields:

- **Object Name.** Enter *MySQL\_version*.
- **SQL Query.** Enter the following:

```
show global variables like 'version'
```

- This query searches an internal table that stores configuration information about the MySQL database (`show global variables`).
  - The query retrieves the value of the status variable "version" (`like 'version'`).
  - The variable "version" contains the version number and version name of the MySQL server.
- **Class Type.** Select *10 Config Character*. The returned value contains both numbers and letters.
  - **Group Number.** Select *Group 1* and *Standard*. In the **Configuration Report** page, this collection object will appear in table 1.
  - **Asset/Form Link.** Leave these fields blank.



- **Inventory Link.** Select *Disabled*.
- **Change Alerting.** Select *Disabled*. SL1 will not trigger an event if the value of this collection object changes. Notice that we will enable this option for all other collection objects in this Dynamic Application. SL1 will then automatically generate events if the value of these collection objects change.
- **Table Alignment.** Select *Left*. This specifies that column-values will be left justified and will not be translated from hexadecimal unicode.
- **Trended Column.** Enter *Value*. The query for this collection object returns two columns: *Variable\_name*, which contains the name of the variable (version) and *Value*, which contains the value of the variable. We are interested only in the value in the *Value* column and want that value to appear in the **Configuration Report** page.
- **Description.** Leave blank.
- **Formula.** Leave blank.

6. Select the **[Save]** button to save the new collection object.

7. For the remaining collection objects, enter the following and select the **[Save]** button for each collection object. If a field is not specified in this table, accept its default value.

Object Name	SQL Query	Class Type	Group Number	Change Alerting	Trended Column
auto_commit	show global variables like 'autocommit'	[15] Config Numeric	2	Enabled	Value
max_connections	show global variables like 'max_connections'	[15] Config Numeric	2	Enabled	Value
key_cache_size	show global variables like 'key_buffer_size'	[15] Config Numeric	3	Enabled	Value
thread_cache_size	show global variables like 'thread_cache_size'	[15] Config Numeric	3	Enabled	Value
query_cache_size	show global variables like 'query_thread_size'	[15] Config Numeric	3	Enabled	Value

---

## Creating a Credential for the MySQL:DBConfiguration Dynamic Application

If you want to align the **MySQL:DBConfiguration** Dynamic Application with SL1's MySQL database, you must create a database credential that allows access to SL1's MySQL database.

For details on creating a credential for SL1's MySQL database, see the [previous section](#).

---

## Aligning the Dynamic Application with a Device


For our example, we aligned the **MySQL:DBConfiguration** Dynamic Application with the Database Server.

There are three ways to align the **MySQL:DBConfiguration** Dynamic Application with a Device:

- **During initial discovery or nightly auto-discovery.** Because the **MySQL:DBConfiguration** Dynamic Application includes a discovery object, SL1 can automatically align this Dynamic Application with devices during discovery. For details on discovery, see the manual **Discovery and Credentials**.
- **Manual discovery for the Dynamic Application.** Because the **MySQL:DBConfiguration** Dynamic Application includes a discovery object, SL1 can automatically align this Dynamic Application with devices. From the **Dynamic Applications Manager** page, you can manually execute discovery for all devices, but only for the **MySQL:DBConfiguration** Dynamic Application. For details on how to perform this type of discovery, see the manual **Dynamic Application Development**.
- **Manually associate the Dynamic Application with an existing device.** For details on how to perform this type of discovery, see the manual **Dynamic Application Development**.

This section will walk you through the steps to manually align the **MySQL:DBConfiguration** Dynamic Application with the Database Server or the All-In-One Appliance.

To manually align the Dynamic Application to the Database Server or the All-In-One Appliance:

1. Go the **Appliance Manager** page (System > Settings > Appliances). Determine the device name of the Database Server or the All-In-One Appliance.
2. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
3. Find the device where the database resides (either the Database Server or the All-In-One Appliance). Select the device's wrench icon (.
4. The **Device Properties** page appears. Select the **[Collections]** tab.
5. In the **Dynamic Application Collections** page, select the **[Action]** menu and select *Add Dynamic Application*.
6. The **Dynamic Application Alignment** page appears. Select **MySQL:DBConfiguration** in the **Dynamic Applications** pane, and select **EM7 DB** in the **Credentials** pane.
7. Select the **[Save]** button to add the Dynamic Application to the device.
8. At the next polling period, SL1 should start collecting the data specified in the **MySQL:DBConfiguration** Dynamic Application from the device where SL1's database resides.

---


## Viewing the Configuration Report for the Dynamic Application

For our example, we aligned the **MySQL:DBConfiguration** Dynamic Application with the Database Server on our device **em7\_ao**.

SL1 automatically creates a Configuration Report on each device that is aligned with the Dynamic Application. The Configuration Report displays the latest data collected from the device by the Dynamic Application.

**NOTE:** For more details on the **Configuration Report** page and a description of the actions you can perform from the **Configuration Report** page, see the manual **Monitoring Device Infrastructure Health**.

To view the Configuration Report for the Dynamic Application:

1. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
2. Find the device that you aligned with the Dynamic Application. (In our example, this device is **em7\_ao**). Select its bar-graph icon (.
3. Select the **[Configs]** tab. The **Configuration Report** page appears.
4. In the NavBar, find the entry for the **MySQL:DBConfiguration** Dynamic Application and expand it.
5. In the **Configuration Report** page, notice how the values are stored in tables as we specified in the **Group** field for each collection object. That is, `mysql_version` is displayed in the first table, `auto_commit` and `max_connections` are displayed in the second table, and `key_cache_size`, `query_cache_size`, and `thread_cache_size` are displayed in the third table.

---

# Example

# 3



## Example of a Dynamic Application with an Identity-Based Relationship

---

### Overview

In this example, we will describe how to build a pair of Dynamic Applications that create relationships between Sciencelogic Database Servers and the Data Collectors and Message Collectors in the same system.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ().
- To view a page containing all of the menu options, click the Advanced menu icon (  ).

This chapter covers the following topics:

<i>Building Dynamic Applications with an Identity-Based Relationship</i> .....	37
<i>Defining the Basic Properties for the Dynamic Application</i> .....	38
<i>Defining the Collection Objects</i> .....	38
<i>Creating the Dynamic Application that Creates the Relationship</i> .....	39
<i>Aligning the Dynamic Applications</i> .....	39
<i>Viewing Dynamic Application Relationships</i> .....	40

---

## Building Dynamic Applications with an Identity-Based Relationship

In this example, we will describe how to build a pair of Dynamic Applications that create relationships between Sciencelogic Database Servers and the Data Collectors and Message Collectors in the same system.

In this example, the data used to build the relationships is in the `master.system_settings_licenses` database table. This table contains information about SL1 Appliances, including the appliance type, name, IP address, etc. On a Database Server, this table contains a row for each appliance in the system. On a Data Collector or Message Collector, this table contains a row for only that Data Collector or Message Collector.

The following database fields are used in this example:

- **id**. The automatically assigned ID number for the appliance. This will be combined with the host name of the appliance to create a unique identifier for the relationships.
- **name**. The host name of the appliance configured in the System > Settings > Appliances page. This will be combined with the id of the appliance to create a unique identifier for the relationships.
- **function**. The type of appliance. The appliance types are stored as integer values. This field contains 5 for Data Collectors and 6 for Message Collectors.

The following database query is used in this example:

```
SELECT CONCAT(CAST(id AS CHAR), name) as uid,  
  
CASE  
  
WHEN function=5 THEN "Data Collector"  
  
WHEN function=6 THEN "Message Collector"  
  
END as func  
  
FROM master.system_settings_licenses  
  
WHERE function IN (5,6);
```

This query returns two values:

- **uid**. A string concatenation of the id and name fields. This will be used as the unique identifier for the relationships. This example assumes that this combination of fields is unique for all appliances in all SL1 systems.
- **func**. A string representation of the appliance type. This will be used as the namespace for the relationships.

The query returns rows only for Data Collectors and Message Collectors.

---

## Defining the Basic Properties for the Dynamic Application

To create the Dynamic Application that creates the identity, which will be aligned to Data Collectors and Message Collectors, perform the following steps:

**NOTE:** For details on each field and its possible options, see the manual *Dynamic Application Development*.

1. Go to the **Dynamic Applications Manager** page (System > Settings > Appliances).
2. Select the **[Actions]** button, and then select **Create New Dynamic Application**. The **Dynamic Applications Create New Application** page appears.
3. Supply values in the following fields:
  - **Application Name**. Enter "ScienceLogic Collector Identity"
  - **Application Type**. Select *Database Configuration*.
  - **Poll Frequency**. To see data as quickly as possible, select *Every 1 Minute* in this field.
4. This example does not have specific requirements for the other settings defined in this page. You can leave the remaining fields set to the default values. Select the **[Save]** button to save the Dynamic Application.

---

## Defining the Collection Objects

To create the collection objects for the Dynamic Application that creates the identity:

**NOTE:** For more details on collection objects, see the manual *Dynamic Application Development*.



1. Select the **[Collections]** tab. The **Dynamic Applications | Collections Objects** page appears.
2. Supply values in the following fields to create the collection object for the *relationship identifier*:
  - **Object Name**. Enter *ID*.
  - **SQL Query**. Enter the database query for this example.
  - **Group**. The namespace and identifier collection objects must be in the same group. Select *Group 1* in this field.
  - **Usage Type**. This collection object will collect the uid values from the database query, which are the identifiers for the relationships. Select *Group Index* in this field.
  - **Trended Column**. This collection object will collect the uid values from the database query. Enter "uid" in this field.
3. Select the **[Save]** button to save the new collection object.
4. Select the **[Reset]** button to clear the form fields.

5. Supply values in the following fields to create the collection object for the **relationship namespace**:
  - **Object Name**. Enter *Appliance Type*.
  - **SQL Query**. Enter the database query for this example.
  - **Group**. The namespace and identifier collection objects must be in the same group. Select *Group 1* in this field.
  - **Usage Type**. This collection object will collect the appliance function values from the database query, which are the namespaces for the relationships. Select *Identity Namespace* in this field.
  - **Trended Column**. This collection object will collect the appliance function values from the database query. Enter "func" in this field.
6. Select the **[Save]** button to save the new collection object.

---

## Creating the Dynamic Application that Creates the Relationship

Both Dynamic Applications in this example collect the same data with only minor configuration differences. Therefore, to create the Dynamic Application that creates the relationship, which will be aligned to Database Servers, you can create a copy of the ScienceLogic Collector Information Dynamic Application as the starting point. To create the Dynamic Application that creates the relationship, perform the following steps:



1. Go to the **Dynamic Applications Manager** page (System > Settings > Appliances).
2. Select the wrench icon () for the "ScienceLogic Collector Identity" Dynamic Application.
3. Supply values in the following fields:
  - **Application Name**. Enter "ScienceLogic Collector Information"
4. Select the **[Save As]** button to save the new Dynamic Application.
5. Select the **[Collections]** tab. The **Dynamic Applications | Collections Objects** page appears.
6. The difference between the collection objects in the identity-defining Dynamic Application (the previous Dynamic Application) and the relationship-defining Dynamic Application (this Dynamic Application) is the **Usage Type** setting for the namespace collection object. Select the wrench icon () for the *Appliance Type* object.
7. In the **Usage Type** drop-down list, select *Relationship Namespace*.
8. Select the **[Save]** button to save the collection object.

---

## Aligning the Dynamic Applications

To align the example Dynamic Applications:

1. Discover the Database Server, Data Collectors, and Message Collectors in a SL1 system.
2. Additionally, you must add a firewall rule to the monitored Data Collectors and Message Collectors to allow connections on port 7707 from the All-In-One Appliance or Data Collector that is performing collection.

3. Align the **ScienceLogic Collector Information** Dynamic Application with the Database Server and use the default Central Database credential. To do this:
4. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
5. Find the Database Server. Select its wrench icon ()
6. Select the **[Collections]** tab.
7. From the **[Actions]** menu, select *Add Dynamic Application*.
8. In the **Dynamic Application Alignment** modal page, select the following:
  - **Dynamic Applications**. Select *ScienceLogic Collector Information*.
  - **Credentials**. Select *EM7 Central Database*. If the monitored Database Server does not use the default username and password, you must edit the credential to use the new username and password.
9. Align the **ScienceLogic Collector Identity** Dynamic Application with the Data Collectors and Message Collectors and use the default Collector Database credential. To do this:
10. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic SL1 user interface).
11. Find the Data Collector. Select its wrench icon ()
12. Select the **[Collections]** tab.
13. From the **[Actions]** menu, select *Add Dynamic Application*.
14. In the **Dynamic Application Alignment** modal page, select the following:
  - **Dynamic Applications**. Select *ScienceLogic Collector Identity*.
  - **Credentials**. Select *EM7 Central Database*. If the monitored Database Server does not use the default username and password, you must edit the credential to use the new username and password.
15. Repeat the bulleted steps for each Data Collector and Message Collector.

---

## Viewing Dynamic Application Relationships

If the example Dynamic Applications are configured correctly and collection is successful, the relationships are displayed in the following places:



- In the **[Topology]** tab in the **Device Reports** panel for the Database Server, Data Collectors, and Message Collectors:

- In the **Device Relationships** page (Registry > Networks > Device Relationships):

Device Relationships | Relationships Found [52104] Trace Reset Guide

	Child *	Child IP	Child Interface *	Child Phys Addr	Child # Manufacture	Parent *	Parent Interface	Parent # Alias	Parent Phys Addr	Parent # Manufacture	Type *
1	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
2	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
3	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
4	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
5	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
6	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
7	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
8	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
9	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
10	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
11	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
12	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
13	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
14	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
15	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
16	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping
17	--	--	--	--	--	10.2.20.6	--	--	--	--	Component Mapping

Viewing Page: 0/1

© 2003 - 2025, ScienceLogic, Inc.

All rights reserved.

#### LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

#### Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

#### Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: [legal@sciencelogic.com](mailto:legal@sciencelogic.com). For more information, see <https://sciencelogic.com/company/legal>.

ScienceLogic

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010