

# Archiving the DataStore

Depending on the amount of disk space available for the continuously growing DataStore, administrators can set an archive policy that determines how many month's worth of data is retained. Old performance data is automatically archived and removed from the DataStore. This archiving procedure works with all databases that are compatible with up.time.

The existing archive policy can be viewed and modified on the **Archive Policy** subpanel, which is accessed from the main **Config** panel. Here, the main archive categories are shown, along with the number of months for which collected data is retained in the DataStore.

Every month, up.time checks the DataStore's entries; data that is older than the limit set in the archive policy are written to XML files. The XML archives use the following format:

```
<table_name>_<date>.xml.gz
```

## Topics on this page

- [Archive Categories](#)
- [Configuring an Archive Policy](#)
- [Restoring Archived Data](#)
- [Exporting and Importing the DataStore](#)

The archives created reflect the database table structure used to store performance data, as well as the date that the stored data represents, as shown in the following example:

```
performance_cpu_2013-09-13.xml.gz
```

The DataStore is trimmed and the XML files are compressed and stored in the `/archives` directory.

For example, if you installed up.time in the default location, the path to the archived data will be:

- Linux: `/usr/local/uptime/archives`
- Windows: `C:\Program Files\uptime software\uptime\archives`

Once backed up, archives can be stored offline. If required, they can be temporarily imported into the DataStore.

## Archive Categories

The following table lists the statistical categories whose archiving can be configured, along with the corresponding DataStore database table:

Archive Policy Category	Database Table
Overall CPU/Memory	performance_cpu
Multi-CPU	performance_aggregate
Detailed Process	performance_psinfo
Disk Performance	performance_disk
File System Capacity	performance_fscap
Network	performance_network
User Information	performance_who
Volume Manager	performance_vxvol
Retained Data	erdc_int_data erdc_decimal_data erdc_string_data

vSphere Performance Data	vmware_perf_aggregate vmware_perf_cluster vmware_perf_datastore_usage vmware_perf_datastore_vm_usage vmware_perf_disk_rate vmware_perf_entitlement vmware_perf_host_cpu vmware_perf_host_disk_io vmware_perf_host_disk_io_adv vmware_perf_host_network vmware_perf_host_power_state vmware_perf_mem vmware_perf_mem_advanced vmware_perf_network_rate vmware_perf_vm_cpu vmware_perf_vm_disk_io vmware_perf_vm_network vmware_perf_vm_power_state vmware_perf_vm_storage_usage vmware_perf_vm_vcpu vmware_perf_watts vsync_update
vSphere Inventory Updates	virtual_inventory_update vmware_event table
Network Device Performance Data	net_device_perf_ping net_device_perf_port

## Configuring an Archive Policy

To set an archive policy, do the following:

1. On the up.time tool bar, click **Config**.
2. In the tree panel, click **Archive Policy**.
3. For the following categories, specify the number of months worth of data that will be retained in the DataStore before being removed and archived:
  - Overall CPU/Memory Statistics
  - Multi-CPU Statistics
  - Detailed Process Statistics
  - Disk Performance Statistics
  - File System Capacity Statistics
  - Network Statistics
  - User Information Statistics
  - Volume Manager Statistics
  - Retained Data
  - vSphere Performance Data
  - vSphere Inventory Data
  - Network Device Performance Data
4. Ensure the **Enable Archiving** checkbox is selected.
5. Click **Set Archive Policy**.
6. Optionally, you can click the **Archive Now** button to immediately create archives of the data in your DataStore. up.time will check the DataStore entries and archiving anything that is older than the limits you have configured.

## Restoring Archived Data

If you need to generate graphs or reports on older data that have already been archived and is no longer in the DataStore, you can import specific archives using the `restorearchive` command line utility. The command's parameters allow you to import archives in the following manner:

- a single archive that represents a specific archive category and date; the collected data for each archive category and 24-hour period is exported to individual XML files
- all archives for a specific date (that is, a 24-hour period)

### Importing Archived Data into the DataStore

To import archived data into the DataStore, do the following:

1. Ensure the database hosting the DataStore is running.
2. At the command line, navigate to the `up.time /scripts` folder.  
For example, if you installed the Monitoring Station in the default location on a Windows system, navigate to the following folder:  
`C:\Program Files\uptime software\uptime\scripts\`
3. Run the `restorearchive` command with one or more of the following options:

Option	Description
<code>-f &lt;filename&gt;</code>	Imports a single file (an archive category's data for a single date). You must specify the full path to the file name.
<code>-d &lt;date&gt;</code>	Imports all files with the specified date (in YYYY-MM-DD format).
<code>-D &lt;directory&gt;</code>	The directory containing the archived files. Note that you must specify this option when using the <code>-d</code> option.
<code>-c &lt;directory&gt;</code>	The full directory path to the <code>uptime.conf</code> file.

For example, the following command would import all of the data archived on September 18, 2013, which are located in the default directory for archived data:

```
restorearchive -d 2006-09-18 -D /usr/local/uptime/archives/ -c /usr/local/uptime
```



If you have deployed up.time UI instances, ensure you always run command-line scripts such as `restorearchive` on the primary Monitoring Station that is actually performing data collection.

## Exporting and Importing the DataStore

In cases where you need to perform a wholesale backup of the existing DataStore (e.g., migrating your DataStore to another database), up.time includes two command-line utilities:

- `fulldatabasedump`: creates a compressed XML file of the contents of your DataStore
- `fulldatabaseimport`: imports the archived data back into your DataStore

Both utilities work with all of the databases supported by up.time.

### Archiving the DataStore

To archive your DataStore, do the following:

1. Ensure the database hosting the DataStore is running.
2. Stop the up.time Data Collector service (i.e., the up.time Core).  
See [Stopping and Restarting up.time Services](#) for more information.
3. At the command line, navigate to the `up.time /scripts` folder.  
For example, if you installed the Monitoring Station in the default location on a Windows system, navigate to the following folder:  
`C:\Program Files\uptime software\uptime\scripts\`
4. Run the following command:  
`fulldatabasedump`



If you have deployed up.time UI instances, ensure you always run command-line scripts such as `fulldatabasedump` or `fulldatabaseimport` on the primary Monitoring Station that is actually performing data collection.

Depending on the size of your DataStore, this process can take anywhere from several minutes to several hours.

The utility creates the file `uptimedump_YYYY-MM-DD.xml.gz` (for example, `uptimedump_2007-01-02.xml.gz`). This file is saved in up.time's root installation directory.

5. Restart the up.time Data Collector service.

## Restoring the DataStore

To restore your DataStore, do the following:

1. Ensure the database hosting the DataStore is running.
2. Use the `resetdb` utility with the `really` and `nodata` options to delete, then recreate the database structure that is used by `up.time` by running one of the following commands:
  - Linux: `/usr/local/uptime/resetdb --nodata really`
  - Windows: `C:\Program Files\uptime software\uptime\resetdb --nodata really`



If you have set up UI instances of `up.time`, ensure you always run command-line scripts such as `resetdb` on the primary Monitoring Station that is actually performing data collection.

3. Run the following command:

```
fulldatabaseimport path/<fileToImport>.xml.gz
```

Where "`path/<fileToImport>.xml.gz`" is the path to, and file name of, the archived contents of your DataStore. For example, to import an archive that is located in `up.time`'s root installation directory, you would enter the following:

```
fulldatabaseimport uptimedump_2007-01-02.xml.gz
```