

# Introduction to datasets

5 minutes

*Datasets* are versioned packaged data objects that can be easily consumed in experiments and pipelines. Datasets are the recommended way to work with data, and are the primary mechanism for advanced Azure Machine Learning capabilities like data labeling and data drift monitoring.

## Types of dataset

Datasets are typically based on files in a datastore, though they can also be based on URLs and other sources. You can create the following types of dataset:

- **Tabular:** The data is read from the dataset as a table. You should use this type of dataset when your data is consistently structured and you want to work with it in common tabular data structures, such as Pandas dataframes.
- **File:** The dataset presents a list of file paths that can be read as though from the file system. Use this type of dataset when your data is unstructured, or when you need to process the data at the file level (for example, to train a convolutional neural network from a set of image files).

## Creating and registering datasets

You can use the visual interface in Azure Machine Learning studio or the Azure Machine Learning SDK to create datasets from individual files or multiple file paths. The paths can include wildcards (for example, `/files/*.csv`) making it possible to encapsulate data from a large number of files in a single dataset.

After you've created a dataset, you can *register* it in the workspace to make it available for use in experiments and data processing pipelines later.

## Creating and registering tabular datasets

To create a tabular dataset using the SDK, use the `from_delimited_files` method of the `Dataset.Tabular` class, like this:

```
Python
```

[Copy](#)

```
from azureml.core import Dataset

blob_ds = ws.get_default_datastore()
csv_paths = [(blob_ds, 'data/files/current_data.csv'),
              (blob_ds, 'data/files/archive/*.csv')]
tab_ds = Dataset.Tabular.from_delimited_files(path=csv_paths)
tab_ds = tab_ds.register(workspace=ws, name='csv_table')
```


The dataset in this example includes data from two file paths within the default datastore:

- The **current\_data.csv** file in the **data/files** folder.
- All **.csv** files in the **data/files/archive/** folder.

After creating the dataset, the code registers it in the workspace with the name **csv\_table**.

## Creating and registering file datasets

To create a file dataset using the SDK, use the **from\_files** method of the **Dataset.File** class, like this:

Python	 Copy
<pre>from azureml.core import Dataset  blob_ds = ws.get_default_datastore() file_ds = Dataset.File.from_files(path=(blob_ds, 'data/files/images/*.jpg')) file_ds = file_ds.register(workspace=ws, name='img_files')</pre>	

The dataset in this example includes all **.jpg** files in the **data/files/images** path within the default datastore:


After creating the dataset, the code registers it in the workspace with the name **img\_files**.

## Retrieving a registered dataset

After registering a dataset, you can retrieve it by using any of the following techniques:

- The **datasets** dictionary attribute of a **Workspace** object.
- The **get\_by\_name** or **get\_by\_id** method of the **Dataset** class.

Both of these techniques are shown in the following code:

Python	 Copy
<pre>import azureml.core from azureml.core import Workspace, Dataset</pre>	

```
# Load the workspace from the saved config file
ws = Workspace.from_config()

# Get a dataset from the workspace datasets collection
ds1 = ws.datasets['csv_table']

# Get a dataset by name from the datasets class
ds2 = Dataset.get_by_name(ws, 'img_files')
```

## Dataset versioning

Datasets can be *versioned*, enabling you to track historical versions of datasets that were used in experiments, and reproduce those experiments with data in the same state.

You can create a new version of a dataset by registering it with the same name as a previously registered dataset and specifying the **create\_new\_version** property:

Python

 Copy

```
img_paths = [(blob_ds, 'data/files/images/*.jpg'),
              (blob_ds, 'data/files/images/*.png')]
file_ds = Dataset.File.from_files(path=img_paths)
file_ds = file_ds.register(workspace=ws, name='img_files',
                           create_new_version=True)
```

In this example, the .png files in the **images** folder have been added to the definition of the **img\_paths** dataset example used in the previous topic.

## Retrieving a specific dataset version

You can retrieve a specific version of a dataset by specifying the **version** parameter in the **get\_by\_name** method of the **Dataset** class.

Python

 Copy

```
img_ds = Dataset.get_by_name(workspace=ws, name='img_files', version=2)
```

## Next unit: Use datasets

Continue >

