

system.dataset.toDataSet

This function is used in **Python Scripting**.

Description

This function is used to 1) convert PyDataSets to DataSets, and 2) create new datasets from raw Python lists. When creating a new dataset, headers should have unique names.

Client Permission Restrictions

This scripting function has no [Client Permission](#) restrictions.

Syntax

system.dataset.toDataSet(dataset)

- Parameters
 - [PyDataSet](#) dataset - A PyDataSet object to convert.
- Returns
 - [DataSet](#) - The newly created dataset.
- Scope
 - All

Syntax

system.dataset.toDataSet(headers, data)

- Parameters
 - [PySequence](#) headers - The column names for the dataset to create.
 - [PySequence](#) data - A list of rows for the new dataset. Each row must have the same length as the headers list, and each value in a column must be the same type.
- Returns
 - [DataSet](#) - The newly created dataset.
- Scope
 - All

Code Examples

Code Snippet

```
# This example create a single column dataset.
header = ['myColumn']
rows = [[1], [2]]
dataset = system.dataset.toDataSet(header, rows)
```

Code Snippet

This first example shows how this function can be used to convert from a PyDataSet (which is what system.db.runQuery returns) to a normal DataSet, which is the datatype of a Table component's data property.

```
pyDataSet = system.db.runQuery("SELECT * FROM example1 LIMIT 100")
table = event.source.parent.getComponent("Table")
normalDataSet = system.dataset.toDataSet(pyDataSet)
table.data = normalDataSet
```

Code Snippet

This second example shows how to use this function to create a new dataset out of a Python sequence that you have filled in. In this case, the sequence is created via a for loop appending rows to a list.

```
# Generate the Rows
rows = []
for x in range(10):
    oneRow = ["Row %d" % x, x+15]
    rows.append(oneRow)

# Generate the DataSet
headers = ["RowID", "Value"]
data = system.dataset.toDataSet(headers, rows)

# Use our new dataset to fill in a Table
table = event.source.parent.getComponent("Table")
table.data = data
```