

Article

[Evgeny Shvarov](#) · Sep 20, 2021 3m read

Importing CSV into the Existing Table in InterSystems IRIS

Hi folks!

Sometimes we need to import data into InterSystems IRIS from CSV. It can be done e.g. via [csvgen](#) tool that generates a class and imports all the data into it.

But what if you already have your own class and want to import data from CSV into your existing table?

There are numerous ways to do that but you can use csvgen (or [csvgen-ui](#)) again! I prepared an example and happy to share. Here we go!

The concept is the following: I have Class A and I want the data in file.csv that contains a column I need for my class.

The steps:

1. create Class B using csvgen,
2. perform SQL Update to add class B data to the class A
3. delete Class B.

To demonstrate a concept I created a [simple demo project](#). The project imports [Countries dataset](#) that contains `dcdata.Country` class with different information on countries including GNP.

```
ClassMethod ImportCSV() As %Status
{
set sc = ##class(communit.csvgen).GenerateFromURL("https://raw.githubusercontent.com/evshvarov/test-ipad/master/gnp.csv", ",", "dc.data.GNP")

Return sc
}
```

But the data on GNP is outdated and I have the recent one in [this CSV](#). Here is the method that shows GNP e.g. for Angola:

```
ClassMethod ShowGNP() As %Status
{
Set sc = $$$OK

&sql(

SELECT TOP 1 name,gnp into :name,:gnp from dc_data.Country
```

```
)

if SQLCODE < 0 throw ##class(%Exception.SQL).CreateFromSQLCODE(SQLCODE,"Show Country
GNP")

write "Country ",name," gnp=",gnp,!

Return sc

}
```

So I import CSV in a generated class with one line:

```
ClassMethod ImportCSV() As %Status

{

set sc = ##class(community.csvgen).GenerateFromURL("https://raw.githubusercontent.com
/evshvarov/test-ipad/master/gnp.csv",",","dc.data.GNP")

Return sc

}
```

and with the second line I perform an SQL query that imports the updated GNP data into my `dc_data.Country` class.

```
ClassMethod UpdateGNP() As %Status

{

Set sc = $$$OK

&sql(

UPDATE dc_data.Country

SET Country.gnp=GNP."2020"

FROM

dc_data.Country Country

INNER JOIN dc_data.GNP GNP

On Country.name=GNP.CountryName

)

if SQLCODE < 0 throw ##class(%Exception.SQL).CreateFromSQLCODE(SQLCODE,"Importing dat
a")
```

```
w "Changes to GNP are made from dc.data.GNP",!
```

```
Return sc
```

```
}
```

And then I delete generated class with its data as I don't need it any more.

```
ClassMethod DropGNP() As %Status
```

```
{
```

```
Set sc = $$$OK
```

```
&sql(
```

```
DROP TABLE dc_data.GNP
```

```
)
```

```
if SQLCODE < 0 throw ##class(%Exception.SQL).CreateFromSQLCODE(SQLCODE,"Drop csv table")
```

```
write "dc.data.DNP class is deleted.",!
```

```
Return sc
```

```
}
```

Well, here is the method that does all at once:

```
ClassMethod RunAll() As %Status
```

```
{
```

```
Set sc = $$$OK
```

```
zw ..ImportDataset()
```

```
zw ..ShowGNP()
```

```
zw ..ImportCSV()
```

```
zw ..UpdateGNP()
```

```
zw ..ShowGNP()
```

```
zw ..DropGNP()
```

```
Return sc
```

```
}
```

Of course it's just one approach to the problem but I hope it can be helpful. Looking forward for your feedback!

[#CSV](#) [#Data Import and Export](#) [#SQL](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/importing-csv-existing-table-intersystems-iris>