
Article

[Robert Cemper](#) · Mar 26, 2019 2m read

[Open Exchange](#)

Synchronize data with DSTIME

Other Sync-Tools just work from Caché/IRIS to Caché/IRIS.

Synchronizing your data to some external DB requires some other solution.

The solution is available in Caché/IRIS since quite some time and works excellent.

^OBJ.DSTIME does the magic.

<https://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=D2IMPchcurrent#D2IMPC23869>

It was built to allow data synchronization with DeepSee.

It keeps a very simple journal on Object/Table changes by signaling Modified, New, Deleted

This could be useful not only for DeepSee but for any other type of Table Synchronization.

The Global *^OBJ.DSTIME* has 2 additional features

- It is wrapped in a persistent class %SYSTEM.DSTIME
<https://docs.intersystems.com/latest/csp/documatic/%25CSP.Documatic.cls?PAGE=CLASS&LIBRARY=%25SYS&CLASSNAME=%25SYSTEM.DSTIME> so you can use it also as normal SQL Table to select your changes
- It maintains a version ID (named DSTIME) that allows control of synchronized junks: + you fetch the last version + increase the version + and then you upload your changes wherever you require them

And as you do the synchronization by pure SQL your target can be *any* DB understanding SQL.

I extended class %SYSTEM.DSTIME to allow Pure SQL Operation

The demo class is a copy of Sample.Person and it runs in namespace IRISAPP.

Typical scenario:

```
IRISAPP>write ##class(OBJ.Person).Populate(15)
15
IRISAPP>do $system.SQL.Shell()
IRISAPP>>DELETE FROM OBJ.PERSON WHERE ID IN (2,5,9)
IRISAPP>>Update OBJ.PERSON SET NAME='Robert' WHERE ID IN (3,7)
```

Now we can take a look on OBJ.DSTIME

```
SAMPLES>>SELECT * FROM OBJ.DSTIME
```

```
DSTIME ClassName ObjectId FilingOp LastVersion Version
0 OBJ.Person 1 1 0 0
0 OBJ.Person 2 2 0 0
0 OBJ.Person 3 0 0 0
0 OBJ.Person 4 1 0 0
0 OBJ.Person 5 2 0 0
```

Synchronize data with DSTIME

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
0 OBJ.Person 6 1 0 0
0 OBJ.Person 7 0 0 0
0 OBJ.Person 8 1 0 0
0 OBJ.Person 9 2 0 0
0 OBJ.Person 10 1 0 0
0 OBJ.Person 11 1 0 0
0 OBJ.Person 12 1 0 0
0 OBJ.Person 13 1 0 0
0 OBJ.Person 14 1 0 0
0 OBJ.Person 15 1 0 0
```

next we set a new version

```
SELECT OBJ.DSTIME_NewVersion()
```

and generate some more Persons

```
SAMPLES>write ##class(OBJ.Person).Populate(15)
```

These new records have a new version.

So you may export easily all changes from the previous version of OBJ.DSTIME during normal operation while any additional changes are logged with the new version.

e.g.

```
INSERT INTO MySQL.Person (name,DOB,SSN)
  select Name,DOB,SSN  from OBJ.Person p
    JOIN OBJ.DSTIME d
    on p.ID = d.ObjectId
   where d.ClassName='OBJ.Person'
      and Version = 0
```

Attention

This solution uses the **actual** content of your Objects / Tables. So if the version of the object log is out of date you may see some newer content if additional changes were applied since.

[GitHub](#)

[#InterSystems IRIS](#)

[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/synchronize-data-dstime>