

Article

[Alexander Koblov](#) · Jul 1, 2019 2m read

Transaction suspencion

It ' s often useful to make changes inside the current transaction, that would not be rolled-back if transaction is rolled-back. For example to do some logging.

This can be achieved by using global that is mapped to temporary database -- IRISTEMP. All globals that start with ^IRIS.Temp* are mapped to IRISTEMP by default. Problem with such approach is that IRISTEMP is cleaned on InterSystems IRIS restart, so this log is lost.

What else you can do is -- suspend transaction temporarily, do the logging, and then resume the same transaction.

Method that suspends/resumes transaction is [\\$System.Process.TransactionsSuspended\(switch\)](#).

If switch is 1 transactions are suspended, if switch is 0 transactions are resumed.

For example:

```
USER>tstart

TL1:USER>set ^a=1

TL1:USER>do $system.Process.TransactionsSuspended(1)

TL1:USER>set ^log($I(^log)) = "we set ^a to 1"

TL1:USER>do $system.Process.TransactionsSuspended(0)

TL1:USER>trollback

USER>zw ^a

USER>zw ^log
^log=1
^log(1)="we set ^a to 1"
```

Notice, in this case changes in ^log global are still journaled (if it is mapped to journaled-database).

Some details are in [Suspending All Current Transactions](#) " section of [Transaction Processing](#) " chapter of documentation.

[#Database Transaction Processing](#) [#ObjectScript](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/transaction-suspencion>