

Príklad synchronizovania skriptov

Skript S1 plní objekty typu [Užívateľská premenná](#) nasledovne:

- *U.Int* - celočíselnou hodnotou
- *U.Text* - textovým vyjadrením hodnoty *U.Int*

```
INT _i

_i := 1
FillNext:
    U.Int := _i
    WAIT
    U.Text := %IToStr(_i)
    WAIT
    _i := _i + 1
GOTO FillNext
END
```

Skript S2 testuje, či je hodnota *U.Int* správne textovo vyjadrená v užívateľskej premennej *U.Text*.

```
INT _i

TestNext:
    _i := %StrToI(U.Text)
    IF _i # U.Int THEN
        MESSAGE "Nie su rovnake" ON WS_BC.HIP
    ENDIF
    GOTO TestNext
END
```

Pri takomto usporiadaní dochádza veľmi často ku kolízii. Operácia naplnenia užívateľských premenných *U.Int* a *U.Text* novými hodnotami je prerušovaná skriptom S2, ktorý v nich nachádza nesprávne hodnoty.

Uvedený problém je možné rieši pomocou akcií [GETACCESS](#) a [RELEASEACCESS](#) nasledovne:

Skript S1

```
INT _i
BOOL _ok
TEXT _regString = "Prístup k U.Int"

_i := 1
FillNext:
    _ok := GETACCESS _regString ; zabezpečím si prístup
    IF _ok THEN
        U.Int := _i
        WAIT
        U.Text := %IToStr(_i)
        WAIT
        RELEASEACCESS _regString ; uvoľním prístup
        _i := _i + 1
    ENDIF
    GOTO FillNext
END
```

Skript S2

```

INT _i
BOOL _ok
TEXT _regString = "Prístup k U.Int"

TestNext:
_ok := GETACCESS _regString ; zabezpečím si prístup
IF _ok THEN
    _i := %StrToI(U.Text)
    IF _i # U.Int THEN
        MESSAGE "Nie su rovnake" ON WS_BC.HIP
    ENDIF
    RELEASEACCESS _regString ; uvoľním prístup
ELSE
    DELAY 100[ms]
ENDIF
GOTO TestNext
END

```

Pri takomto usporiadaní skript S2 prístupuje k hodnotám objektov *U.Int* a *U.Text* len v prípade, že sa mu podarila registrácia textu akciou [GETACCESS](#). Úspešná registrácia znamená, že skript S1 nemení hodnoty týchto objektov. Ide o konkurenčný prístup dvoch paralelne bežiacich eventov k rovnakým objektom.



Súvisiace stránky:

[Akcie v skriptoch](#)