ESL Interface

Objekt **ESL Interface** predstavuje prostriedok, ktorého používanie sprehadní a zviditení väzby medzi ESL skriptami v aplikácii (rodi objektu typu ESL Interface je systémový objekt ESLINTERFACES).
Väzby medzi skriptami vznikajú volaním **RPC** alebo **RPCX** procedúr. Tieto volania identifikujú meno procedúry v cieovom skripte textovým reazcom, ktorý sa poas ukladania skriptu nekontroluje. Tým pádom je možné vytvori a uloži skript, ktorý obsahuje volanie neexistujúcej RPC procedúry.

Objekt ESL Interface predstavuje definíciu hlaviky jednej alebo viacerých ESL procedúr bez tela. Formálne teda definuje množinu RPC procedúr. Meno objektu typu ESL Interface môže by použité pri definícii ESL skriptu za kúovým slovom IMPLEMENTATION.

Napríklad:

IMPLEMENTATION I.MsgServer

Za kúovým slovom IMPLEMENTATION musí nasledova meno minimálne jedného objektu typu ESL Interface. Pred ním nesmie by žiadna akcia (definícia procedúry, premennej alebo iná akcia).
ESL skript musí implementova všetky procedúry každého ESL Interface, ktorý je uvedený za kúovým slovom IMPLEMENTATION (hovoríme, že ESL skript **implementuje rozhranie**).
Z opanej strany je možné vola procedúry ESL skriptu, ktorý implementuje nejaké rozhranie práve prostredníctvom mena príslušného rozhrania. Toto umožuje formalizova (a v neposlednom rade sprehadni) logické väzby medzi ESL skriptami, ktoré vznikajú vzájomným volaním RPC procedúr. Význam rozhrania je taký, že predpisuje zoznam procedúr, ktoré musí implementova uritý ESL skript - teda v konenom dôsledku aj implementuje.

Nasledujúci príklad rámcovo demonštruje využitie ESL Interface pri návrhu jednoduchej aplikanej vlastnosti - posielanie správ medzi užívatemi. Táto innos bude na strane užívatea realizovaná jednoduchou schémou **S.MSGClient** a na strane servera objektom typu **Event E.MSGServer.**

Z implementaného hadiska je **E.MSGServer** server, ktorý ponúka urité služby. Tieto služby sú deklarované práve ESL Interface **I.MsgServer** a zaruuje nám existenciu (resp. implementáciu) všetkých potrebných procedúr (deklarovaných na úrovni rozhrania). Rozhranie **I.MsgServer** deklaruje procedúru **RegisterClient**, ktorou sa každý klient, ktorý sa chce zúastni na posielaní správ, registruje a získa svoj jednoznaný identifikátor. Vzhadom na to, že primárna požiadavka je príjem správ klientom (registrovaným) asynchrónne, toto nám zabezpeí **E.MSGServer**, ktorý implementuje procedúru **SendMessage**, interne volajúcu registrovaného klienta, ktorý musí implementova rozhranie **I.MsgClient**. Ten zabezpeí prítomnos procedúry **ReceivMessage**.

Rozhrania, ktoré musia jednotlivé skripty implementova:

Pre **E.MSGServer** to bude rozhranie s menom **I.MsgServer**:

```
 ;*****************************************************************************
 ; Meno objektu: I.MsgServer
 ; Rozhranie MSG Servera
 ; Nasledujuce procedury musi kazdy MSG Server implementovat

 ; Povinna registracia klienta
 ; Registracia prideli jednoznacne _clientUID
 RPC PROCEDURE RegisterClient(IN TEXT _clientName, IN INT _clientHOBJ, IN INT _clientProcessHOBJ, INT
_clientUID)
 ; Odregistrovanie klienta
 RPC PROCEDURE UnRegistrateClient(INT _clientUID)

 ; Procedura vrati zoznam vsetkych registrovanych klientov
 RPC PROCEDURE  GetClientList(RECORD NOALIAS (SD.Public_ClientList) _clients)

 ; Procedura posle spravu _msg registrovanemu klientovi _dstClientUID
 ; Navratovy kod _retCode
 ;   0 - OK
 ;   1 - klient _dstClientUID nie je registrovany
 RPC PROCEDURE SendMessage(IN INT _dstClientUID, IN TEXT _msg, INT _retCode)
 ;*****************************************************************************
```

Minimálna (samozrejme nefunkná) implementácia rozhrania v rámci ESL skriptu je nasledovná:

```
 ; Rozhranie MSG Servera
 IMPLEMENTATION I.MsgServer

 ; Registracia klienta
 IMPLEMENTATION RPC PROCEDURE I.MsgServer^RegisterClient(IN TEXT _clientName, IN INT _clientHOBJ, IN INT
_clientProcessHOBJ, INT _clientUID)
 END RegisterClient

 ; Odregistrovanie klienta
 IMPLEMENTATION RPC PROCEDURE I.MsgServer^UnRegistrateClient(INT _clientUID)
 END UnRegistrateClient

 ; Procedura vrati zoznam vsetkych registrovanych klientov
 IMPLEMENTATION RPC PROCEDURE I.MsgServer^GetClientList(RECORD NOALIAS (SD.Public_ClientList) _clients)
 END GetClientList

 ; Procedura posle spravu _msg registrovanemu klientovi _dstClientUID
 ; Navratovy kod _retCode
 ;   0 - OK
 ;   1 - klient _dstClientUID nie je registrovany
 IMPLEMENTATION RPC PROCEDURE I.MsgServer^SendMessage(IN INT _srcClientUID, _dstClientUID, IN TEXT  _msg, INT
_retCode)
 END SendMessage
```

Implementácia procedúry rozhrania sa zaína kúovým slovom IMPLEMENTATION a jej meno je uvedené menom rozhrania a znakom '^'. ESL skript musí implementova všetky (rozhraním) predpísané procedúry.

Pre **S.MSGClient** to bude rozhranie s menom **I.MsgClient**:

```
;*****************************************************************************
 ; Meno objektu: I.MsgClient
 ; Rozhranie MSG Clienta
 ; Nasledujuce procedury musi kazdy MSG Client implementovat

 ; Prijem spravy _msg od klienta _srcClientUID
 RPC PROCEDURE ReceivMessage(IN INT _srcClientUID, IN TEXT _msg)
 ;*****************************************************************************
```

Z pohadu servera (**E.MSGServer**) nie je podstatné, kto s ním komunikuje (i schéma S.MSGClient, alebo iný objekt), ale dôležité je, i implementuje potrebné rozhranie. Preto v reálnej aplikácii môže by klient reprezentovaný aj inou schémou (alebo objektom typu Event). Dôležité je, aby implementoval rozhranie **I.MsgClient**.

Súvisiace stránky:

Konfigurácia ESL Interface
Akcia PROCEDURE
Akcia IMPLEMENTATION
Doplnenie procedúr v ESL editore