

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 vytvoriť a uložiť 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 byť použité pri definícii ESL skriptu za kúovým slovom IMPLEMENTATION.

Napríklad:

IMPLEMENTATION I.MsgServer

Za kúovým slovom IMPLEMENTATION musí nasledovať meno minimálne jedného objektu typu ESL Interface. Pred ním nesmie byť žiadna akcia (definícia procedúry, premennej alebo iná akcia).

ESL skript musí implementovať všetky procedúry každého ESL Interface, ktorý je uvedený za kúovým slovom IMPLEMENTATION (hovoríme, že ESL skript **implementuje rozhranie**).

Z opanej strany je možné volať procedúry ESL skriptu, ktorý implementuje nejaké rozhranie práve prostredníctvom mena príslušného rozhrania. Toto umožňuje formalizovať (a v neposlednom rade sprehadniť) logické väzby medzi ESL skriptami, ktoré vznikajú vzájomným volaním RPC procedúr. Význam rozhrania je taký, že predpisuje zoznam procedúr, ktoré musí implementovať ESL skript - teda v konečnom dôsledku aj implementuje.

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

Z implementovaného hadiska je **E.MSGServer** server, ktorý ponúka určité služby. Tieto služby sú deklarované práve ESL Interface **I.MsgServer** a zaručuje nám existenciu (resp. implementáciu) všetkých potrebných procedúr (deklarovaných na úrovni rozhrania). Rozhranie **I.MsgServer** deklaruje procedúru **RegisterClient**, ktorou sa každý klient, ktorý sa chce zúčastniť na posielaní správ, registruje a získava svoj jednoznačný identifikátor. Vzhľadom na to, že primárna požiadavka je príjem správ klientom (registrovaným) asynchrónne, toto nám zabezpečí **E.MSGServer**, ktorý implementuje procedúru **SendMessage**, interne volajúcu registrovaného klienta, ktorý musí implementovať rozhranie **I.MsgClient**. Ten zabezpečí prítomnosť procedúry **ReceiveMessage**.

Rozhrania, ktoré musia jednotlivé skripty implementovať:

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

```
;*****  
; Meno objektu: I.MsgServer  
; Rozhranie MSG Servera  
; Nasledujúce procedúry musí každý MSG Server implementovať  
  
; Povinná registrácia klienta  
; Registrácia prideli jednoznačne _clientUID  
RPC PROCEDURE RegisterClient(IN TEXT _clientName, IN INT _clientHOBJ, IN INT _clientProcessHOBJ, INT  
_clientUID)  
; Odregistrovanie klienta  
RPC PROCEDURE UnRegisterClient(INT _clientUID)  
  
; Procedúra vráti zoznam všetkých registrovaných klientov  
RPC PROCEDURE GetClientList(RECORD NOALIAS (SD.Public_ClientList) _clients)  
  
; Procedúra pošle správu _msg registrovanému klientovi _dstClientUID  
; Navratový kód _retCode  
; 0 - OK  
; 1 - klient _dstClientUID nie je registrovaný  
RPC PROCEDURE SendMessage(IN INT _dstClientUID, IN TEXT _msg, INT _retCode)  
;*****
```

Minimálna (samozrejme nefunkčná) 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^UnRegisterClient(INT _clientUID)
END UnRegisterClient

; Procedura vrati zoznam vsetkych registrovanых 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 začína kúovým slovom IMPLEMENTATION a jej meno je uvedené menom rozhrania a znakom '^'. ESL skript musí implementovať všetky (rozhraním) predpísané procedúry.

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

```

;*****
; Meno objektu: I.MsgClient
; Rozhranie MSG Clienta
; Nasledujúce procedúry musí každý MSG Client implementovať

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

```

Z pohľadu 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 byť 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](#)