[Q]: Как вызывать рекс-функции из своей программы?

[A]: Dmitry Zavalishin (2:5020/32)

Это - кусок кода, наспех выдраный из U1 - вряд ли скомпилится у вас, но как пример - сойдет, надеюсь.

#define INCL_REXXSAA #include <rexxsaa.h> /* needed for RexxStart() */ #include <stdio.h> /* needed for printf() */ #include <string.h> /* needed for strlen() */

bool CallRexx(const char *prog, string &out, const char *a1, const char *a2)

```
  {
  RXSTRING arg[2];                    // argument string for REXX
  RXSTRING rexxretval;                // return value from REXX
```

```
  APIRET   rc;                        // return code from REXX
  SHORT    rexxrc = 0;                // return code from function
```

```
  if( prog == NULL || strlen(prog) == 0 )
return Err;
```

```
  /* By setting the strlength of the output RXSTRING to zero, we   */
  /* force the interpreter to allocate memory and return it to us. */
  /* We could provide a buffer for the interpreter to use instead. */
  rexxretval.strlength = 0L;          /* initialize return to empty*/
```

```
  if( a1 == NULL ) a1 = "";
  MAKERXSTRING(arg[0], a1, strlen(a1)); /* create input argument     */
```

```
  if( a2 == NULL ) a2 = "";
  MAKERXSTRING(arg[1], a2, strlen(a2)); /* create input argument     */
```

```
  /* Here we call the interpreter.  We don't really need to use    */
  /* all the casts in this call; they just help illustrate         */
  /* the data types used.                                          */
```

```
  rc=RexxStart((LONG)      2,             /* number of arguments   */
              (PRXSTRING)  &arg,          /* array of arguments    */
              (PSZ)        prog,          /* name of REXX file     */
              (PRXSTRING)  0,             /* No INSTORE used       */
              (PSZ)        "U1",         /* Command env. name     */
              (LONG)       RXSUBROUTINE,  /* Code for how invoked  */
              (PRXSYSEXIT) 0,             /* No EXITs on this call */
              (PSHORT)     &rexxrc,       /* Rexx program output   */
              (PRXSTRING)  &rexxretval ); /* Rexx program output   */
```

debug(“CallRexx() = '%s',int=%d, rexx=%d”,rexxretval.strptr, rc, (int)rexxrc);

printf(“Interpreter Return Code: %d\n”, rc); printf(“Function Return Code: %d\n”, (int) rexxrc); *printf(“Args: '%s', '%s'\n”, arg[0].strptr, arg[1].strptr);* printf(“Ret : '%s'\n”, rexxretval.strptr);

```
  if( rexxretval.strptr != NULL )
      out = rexxretval.strptr;
```

```
  DosFreeMem(rexxretval.strptr);          /* Release storage       */
                                          /* given to us by REXX.  */
```

```
  if( rexxrc != 0 )
{
      error( EI_None, "CallRexx( '%s', out, '%s', '%s' ) returned %d",
    prog, a1, a2, (int) rexxrc );
return Err;
}
```

```
  return rc == 0 ? Ok : Err;
  }
```

From:
https://ftp.osfree.org/doku/ - **osFree wiki**

Permanent link:
https://ftp.osfree.org/doku/doku.php?id=ru:os2faq:os2prog:os2prog.027

Last update: **2014/06/20 05:08**