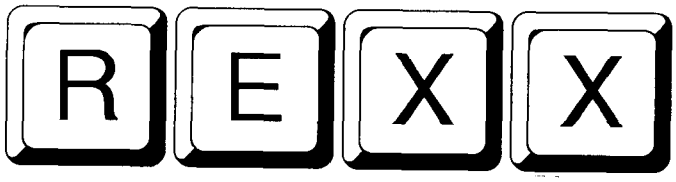
REXX Changes in OS/2 Warp

Dick Goran
CFS Nevada, Inc.

Pages 274-282



Changes in OS/2 Warp

REXX Symposium
May 1-3, 1995
Palo Alto, California

Dick Goran
C F S Nevada, Inc.
953 E. Sahara Avenue, Suite 9B
Las Vegas, Nevada 89104-3012
Voice: 702-732-9616
FAX: 702-732-3847
Email: 71154.2002@CompuServe.com

Warp and REXX

Warp, or OS/2 Warp Version 3.0 as it is officially named, has added 5 new functions to the REXXUTIL application programming interface, or API as IBM likes to call it. Figures 1 through 5 contain a description of these new functions as they appear in the *REXX Reference Summary Handbook*. Unfortunately, IBM did not do a very good job in documenting these new functions.

While copying, moving, or creating shadows of workplace shell objects (WPS) is somewhat intuitive, saving and opening WPS objects with the new functions is not.

SysCopyObject(), **SysMoveObject()**, and **SysCreateShadow()** are fairly straightforward in their purpose. These functions permit a simple means of copying, moving, or creating a shadow of WPS objects from within a REXX program. However, there are some points of special interest when using these three functions.

When an object is copied, no object ID is provided for in the copy. Whether the original object has an object ID or not, the copy will not have an object ID. The only currently available mechanism for assigning an object ID to the copy is with a third-party utility such as ***DeskMan/2***. When a shadow is created, the shadow ID of the newly created object will have the same value as the object ID of the original object.

The purpose of the **SysSaveObject()** function is to force OS/2 to flush the file system objects properties (stored as extended attributes) and the Workplace Shell abstract objects properties (stored in the OS2.INI and OS2SYS.INI files) to disk.

The **SysOpenObject()** function is just as obscurely documented in the online REXX.INF file that comes with Warp. As with many of the other WPS functions contained in REXXUTIL, the IBM-supplied documentation refers to WPS and program manger C^{++} language functions that the average OS/2 user would not have access to without owning the toolkits made for OS/2. The information shown in Figure 1 was compiled from a combination of "bit-digging" research along with some assistance from IBM's Glendale Laboratories - the group responsible for REXX development. The numeric values shown in Figure 1, and used to tell the **SysOpen()** function which view is to be opened, may not be complete. It will take some trial-and-error testing along with independent research to determine what other values may be used. I suggest that users who want to keep up with the latest information, as it becomes available, stay current with the material in the various REXX related fora on CompuServe (OS2DF1, Section 6), IBM's IBMLink and TALKLink (OS2REXX CFORUM), *comp.lang.rexx*on the Internet along with your favorite local BBS.

Development Technologies and Greg Czaja have released version 1.51 of DeskMan/2 with updated REXX functionality as well as interfacing with the WPS functions for

Warp. C F S Nevada, Inc. has released the third edition of the *REXX Reference Summary Handbook* (ISBN 0-9639854-2-6 / IBM SRL & PUBORDER S246-0078-01) with the Warp additions.

One of the other major changes in Warp that is directly related to REXX is the ability to both create and change printer objects (WPPrinter class) and the new LaunchPad (WPLaunchPad) with the REXXUTIL functions. Figure 11 is an example of a REXX program used to replace an existing LaunchPad with one configured within the program.

```
SysSaveObject( object_name, timing_flag )
```

Returns 1 if the WPS object *object_name* was successfully saved; otherwise, returns 0. File system objects (WPFileSystem) are saved in the file system's extended attributes and abstract objects are saved in the OS2.INI (user) file. Transient objects (WPTransient) cannot be saved.

Object_name can be a WPS object ID (the unique string preceded with a '<' and terminated with a '>') assigned to the object when it was created (e.g. <WP_DESKTOP>) or a fully qualified file name.

Timing_flag can be **0** (Boolean false - object is to be saved synchronously) or **1** (Boolean true - object is to saved asynchronously).

Figure 1 - SysSaveObject() function

```
SysCreateShadow( object_name, →
                          → object_destination )
```

Returns 1 if a shadow of *object_name* was successfully created at the specified location, *object_destination*; otherwise, returns 0.

Both *object_name* and *object_destination* can be a WPS object ID (the unique string preceded with a '<' and terminated with a '>') assigned to the object when it was created (e.g. <WP_DESKTOP>) or a fully qualified file name.

Figure 2 - SysCreateShadow() function

```
SysOpenObject( object_name, view, flag )
```

Returns 1 if the WPS object *object_name* was successfully opened on the Desktop; otherwise, returns 0.

Object_name can be a WPS object ID (the unique string preceded with a '<' and terminated with a '>') assigned to the object when it was created (e.g. <WP_DESKTOP>) or a fully qualified file name.

View specifies the view to be opened and can contain either a numeric value or the equivalent string. The function will pass all numeric values to the underlying wpOpen() or wpViewObject() function without testing the value for validity.

0 - DEFAULT
1 - ICON
2 - SETTINGS
3 - HELP
4 - RUNNING
5 - PROMPTDLG
121 - PALETTE

Flag can contain a 1 indicating that an existing view of an object can be opened on top of the Desktop (resurfaced) by calling the wpViewObject method or a 0 indicating that the view specified in *view* is to be opened using the wpOpen method. The following comment originated in the description of the wpOpen method:

"In general, wpViewObject should be used instead of the wpOpen method. This is because wpViewObject takes into consideration the setting in the Object Open Behavior field on the Window page of the Settings notebook for the object. If a view of the object is already open, wpViewObject will depending on the setting of the Object Open Behavior field, either display the existing window for the object or create a new object."

"In contrast, wpOpen always opens a new view of the object. Under certain circumstances this might be called for, but, under most circumstances, wpViewObject should be called instead."

Figure 3 - SysOpenObject() function

```
SysMoveObject( object_name, →
                          → object_destination )
```

Returns 1 if *object_name* was successfully moved to *object_destination*; otherwise, returns 0. If the object already exists in the destination location, it is not moved and a 0 is returned.

Both *object_name* and *object_destination* can be a WPS object ID (the unique string preceded with a '<' and terminated with a '>') assigned to the object when it was created (e.g. <WP_DESKTOP>) or a fully qualified file name.

Figure 4 - SysMoveObject() function

```
SysCopyObject( object_name, →
                        → object_destination )
```

Returns 1 if *object_name* was successfully copied to *object_destination*; otherwise, returns 0. If the object already exists in the destination location, it is not copied and a 0 is returned.

Both *object_name* and *object_destination* can be a WPS object ID (the unique string preceded with a '<' and terminated with a '>') assigned to the object when it was created (e.g. <WP_DESKTOP>) or a fully qualified file name.

Note 01: *The copied object will not have an OBJECTID whether the original object had one assigned or not.*

Note 02: *Some of the object's other properties are not copied along with the object. Specifically, ASSOCTYPE= belonging to the original object does not appear on the copy. This is consistent with what occurs when using drag & drop to copy an object.*

Figure 5 - SysCopyObject() function

Tips on Using REXX and the Workplace Shell

Any changes which are made to an open Settings notebook via **SysSetObjectData()** are not necessarily reflected in that notebook until it is closed and reopened.

If the same key name is specified more than once within a setup string, it generally appears as though the first key name-value pair is the one which prevails; however, that is not always the case.

Where a numeric value of 0 or 1 is used to represent NO or YES respectively; it appears that any numeric value other than 0 will be used as if the value had been 1.

Some of the alphabetic values of the key name=value pairs have been found to be case sensitive with uppercase being required; therefore, all alphabetic values should be created in uppercase.

A new line character, '0A'x, may be used to cause a value such as Title to occupy more than one line. Also, it appears that the occurrence of the escape character, ^, causes a new line to be created; however, 2nd and subsequent escape characters used for this purpose appear to be ignored.

If both ICONFILE and ICONRESOURCE are specified in the same setup string, ICONFILE prevails.

An OBJECTID should not be assigned to an object defined as a template since this would lead to multiple objects with the same OBJECTID.

The object pointer or handle can only be retrieved via the **wpclsQueryObject** method or the **WinQueryObject** function, respectively (neither of which are currently available via REXX).

Prior to Warp, there was no method for altering the background characteristics for a folder other than the bitmap image name (e.g. image vs. color; normal, scaled or tiled image; etc.) using either **SysCreateObject()** or **SysSetObjectData()**. Warp allows all of the characteristics of the Desktop background to be specified.

Prior to Warp, there was no method for altering "Always maintain sort order" using either SysCreateObject() or SysSetObjectData(). Warp introduced the "ALWAYSSORT=YES;" setup string parameter.

If OPEN=SETTINGS is specified, the

program object's notebook is opened; however, if OPEN=DEFAULT is specified, the program object is launched (its icon is cross-hatched) and the program appears in the task list but it does not come to the foreground without either a second call to **SysSetObjectData()** or manual intervention.

Warp / REXX Mystery Failures

There has been a "fix" in Warp Version 3 implemented by IBM that is subtly causing REXX programs, that ran with prior versions of OS/2, to fail. The culprit is the lack of file handles in the OS/2 session where the REXX program is running.

The default number of file handles, a resource required for each open file, is, and has been, twenty. Of the twenty, fifteen are available for user programs with five being reserved for system-related files. Prior to Warp Version 3.0, when multimedia support was installed it changed the default for the number of file handles from twenty to eighty. Apparently, this was being done without the knowledge of the kernel developers and they deemed it necessary to "correct" this problem.

The end result is that if you have a REXX program that inadvertently references file with any of the input/output (I/O) functions: CHARIN(), CHAROUT(), CHARS(), LINEIN(), LINEOUT(), or LINES() or uses any library functions that do not close all of their files (for example - the SysGetMessage() function); these programs may begin to fail. The only way to prevent this from happening is to increase the number of file handles available in the particular session where the program is running. This can be done with the GrowHandles() C function. Quercus Systems has implemented this capability in their latest version of REXXLIB, a commercial product that is available in a fully functional "demo" form from your favorite OS/2 BBS or repository. With the addition of REXXLIB.DLL, you can call the DOSFILEHANDLES() function and specify the number of file handles you want to be available for that session. Once the number of file handles has been increased, the larger number of file handles remain available to that session until the session is closed.

```
/* 9506LS07.CMD (RXLS05.CMD) - Build your own Launchpad */
                                                        /* 0002 */
LaunchPadID = '<WP_LAUNCHPAD>'                          /* 0003 */
location    = '<WP_OS2SYS>'                             /* 0004 */
title       = 'LaunchPad'                               /* 0005 */
class       = 'WPLaunchPad'                             /* 0006 */
                                                        /* 0007 */
/*------------------------*\                            /* 0008 */
|  Setup LaunchPad string  |                            /* 0009 */
\*------------------------*/                            /* 0010 */
parameters =,                                           /* 0011 */
   'CCVIEW=NO;'                                  ||,    /* 0012 */
```

```
      'HELPPANEL=32253;'                                  ||,    /* 0013 */
      'ICONRESOURCE=74 PMWP.DLL;'                         ||,    /* 0014 */
      'LPACTIONSTYLE=OFF;'                                ||,    /* 0015 */
      'LPCLOSEDRAWER=YES;'                                ||,    /* 0016 */
      'LPDRAWERTEXT=YES;'                                 ||,    /* 0017 */
      'LPFLOAT=NO;'                                       ||,    /* 0018 */
      'LPHIDECTLS=YES;'                                   ||,    /* 0019 */
      'LPSMALLICONS=YES;'                                 ||,    /* 0020 */
      'LPTEXT=YES;'                                       ||,    /* 0021 */
      'LPVERTICAL=YES;'                                   ||,    /* 0022 */
      'NOPRINT=YES;'                                      ||,    /* 0023 */
      'FPOBJECTS='                                        ||,    /* 0024 */
         '<WP_DRIVES>,'                  /* 01 */         ||,    /* 0025 */
         '<WPPO_HPLaserJ>,'              /* 02 */         ||,    /* 0026 */
         '<Corel_Draw!>,'                /* 03 */         ||,    /* 0027 */
         '<WP_MEGASCAN^3>,'              /* 04 */         ||,    /* 0028 */
         '<WP_OS2WIN>,'                  /* 05 */         ||,    /* 0029 */
         '<TAPCIS>,'                     /* 06 */         ||,    /* 0030 */
         '<WordPerfect_for_Windows>,'    /* 07 */         ||,    /* 0031 */
         '<WP_GAMES>,'                   /* 08 */         ||,    /* 0032 */
         ';'                                              ||,    /* 0033 */
      'OBJECTID=' || LaunchPadID || ';'                           /* 0034 */
                                                                  /* 0035 */
   call SysCreateObject class,,                                   /* 0036 */
                        title,,                                   /* 0037 */
                        location,,                                /* 0038 */
                        parameters,,                              /* 0039 */
                        'R'                                       /* 0040 */
   if RESULT <> 1 then                                            /* 0041 */
      do                                                          /* 0042 */
         say '   Error creating launchpad'                        /* 0043 */
         exit                                                     /* 0044 */
      end                                                         /* 0045 */
                                                                  /* 0046 */
   /*----------------------*\                                     /* 0047 */
   |  Setup drawer strings  |                                    /* 0048 */
   \*----------------------*/                                     /* 0049 */
   drawer_01 =,                                                   /* 0050 */
      'DRAWEROBJECTS=01,'                                 ||,    /* 0051 */
         '<BackMaster>,'                                  ||,    /* 0052 */
         'L:\WWW,'                                        ||,    /* 0053 */
         ';'                                                      /* 0054 */
                                                                  /* 0055 */
   drawer_02 =,                                                   /* 0056 */
      'DRAWEROBJECTS=02,'                                 ||,    /* 0057 */
         '<WPPO_FxPrint>,'                                ||,    /* 0058 */
         ';'                                                      /* 0059 */
                                                                  /* 0060 */
   drawer_03 =,                                                   /* 0061 */
      'DRAWEROBJECTS=03,'                                 ||,    /* 0062 */
         'c:\os2addon\pmcamera.exe,'                      ||,    /* 0063 */
         ';'                                                      /* 0064 */
```

```
                                                                  /* 0065 */
drawer_05 =,                                                      /* 0066 */
   'DRAWEROBJECTS=05,'                                    ||,     /* 0067 */
      '<WP_WINFS>,'                                       ||,     /* 0068 */
      '<WP_WIN2WIN>,'                                     ||,     /* 0069 */
      '<WP_DOSFS>,'                                       ||,     /* 0070 */
      '<WP_DOSWIN>,'                                      ||,     /* 0071 */
      '<WP_OS2FS>,'                                       ||,     /* 0072 */
      ';'                                                         /* 0073 */
                                                                  /* 0074 */
drawer_06 =,                                                      /* 0075 */
   'DRAWEROBJECTS=06,'                                    ||,     /* 0076 */
      '<IAK_SLIPPM>,'                                     ||,     /* 0077 */
      '<ADV_DIALER>,'                                     ||,     /* 0078 */
      '<WP_OS/2_CIM__>,'                                  ||,     /* 0079 */
      '<WP_INTERNET^C>,'                                  ||,     /* 0080 */
      '<WP_XTALK_^MK_>,'                                  ||,     /* 0081 */
      ';'                                                         /* 0082 */
                                                                  /* 0083 */
drawer_07 =,                                                      /* 0084 */
   'DRAWEROBJECTS=07,'                                    ||,     /* 0085 */
      '<WP_WP_5.1>,'                                      ||,     /* 0086 */
      '<WP_REXX_^HAND>,'                                  ||,     /* 0087 */
      ';'                                                         /* 0088 */
                                                                  /* 0089 */
/*-------------------------------------------------*\             /* 0090 */
|  Add drawers to LaunchPad & open it on Desktop  |              /* 0091 */
\*-------------------------------------------------*/             /* 0092 */
call SysSetObjectData LaunchpadID, drawer_01                      /* 0093 */
call SysSetObjectData LaunchpadID, drawer_02                      /* 0094 */
call SysSetObjectData LaunchpadID, drawer_03                      /* 0095 */
call SysSetObjectData LaunchpadID, drawer_05                      /* 0096 */
call SysSetObjectData LaunchpadID, drawer_06                      /* 0097 */
call SysSetObjectData LaunchpadID, drawer_07                      /* 0098 */
                                                                  /* 0099 */
call SysOpenObject LaunchpadID, 0, 1                              /* 0100 */
exit                                                              /* 0101 */
```

Referenced Resources

REXXLIB - OS/2 REXX API ($20.00 to $50.00)

Quercus Systems
14500 Big Basin Way, Suite E
Saratoga, CA 95070
800-440-5944 orders
408-867-7399 voice
408-867-7489 FAX
408-867-7488 BBS
CompuServe, PCVENA, Sec 11 (GO CIS:QUERCUS - Charles Daney 75300,2450)

REXX Reference Summary Handbook ($27.95) by Dick Goran

C F S Nevada, Inc.
953 E. Sahara Ave, Suite 9B
Las Vegas, Nevada 89104-3012
800-739-9672 orders
702-732-9616 voice
702-732-3847 FAX

Biographical info - Dick Goran:

A veteran of the computer industry for 34 years, Goran is a contributing editor and monthly columnist for *OS/2 Magazine* and serves as one of IBM's OS/2 Advisors on CompuServe. Considered one of the leading authorities on OS/2 REXX, Goran authored the best-selling, award-winning *REXX Reference Summary Handbook*.

His company, C F S Nevada, Inc. located in Las Vegas, offers the OS/2 REXX class to the public as well as publishing the *REXX Reference Summary Handbook*. Goran speaks to OS/2 User Groups and other industry associations throughout the country on both OS/2 and the REXX programming language.

Goran returned to the software business in 1991 after having sold his IBM mainframe systems software development business in 1987, retiring, and relocating to Las Vegas from Boston. While in Las Vegas, Goran began hosting an evening radio talk show and has since appeared in several movies.

Goran is highly visible in the OS/2 forums on CompuServe and can be reached via e-mail at *71154.2002@CompuServe.com*. He also maintains an FTP directory at *ftp.netcom.com:/pub/dg/dgoran* where many of his OS/2 REXX utilities are available to the public at no charge.