ORACLE

Oracle® Documaker

Remote Access

User Guide

12.7.0
Part number: F51808-01
December 2021

ORACLE

Contents

Using the Remote Access Library

The Remote Access Library (RACLib) was created to give non-Oracle Insurance applications the ability to start, stop, and control (to some degree) the Documaker Workstation, also known as the Processing System and PPS. The library provides API functions that can be called from any computer language that can interface to C functions in a DLL. In addition, an ActiveX component (RacCo) is provided for Windows.

This document discusses using Remote Access in one of two ways: through the RacCo ActiveX component or through direct API function calls. The various API functions provided by RACLib are described in detail in this manual as a Function Reference. The RacCo library is detailed in the ActiveX Object Reference and the Class Reference. Topics covered include using C, C++, Visual Basic, and Oracle forms to remotely access the Documaker Workstation.

Verifying Access to the System

Run Documaker Workstation

Before beginning to use the Remote Access Library to customize Documaker Workstation, make sure Documaker Workstation works in your environment. You should be able to run the *AFEMNW32.EXE* program to access your archives and do forms entry (if needed in your environment). For more information on setting up the system, refer to the Documaker Workstation Administration Guide and the Documaker Workstation User Guide.

Path statement

When you write your own application to access Documaker Workstation your program will need to locate and load DLLs and INI files that Documaker Workstation uses. Your application will make calls into RACLib, which will then call into other DLL files of Documaker Workstation. Therefore, the PATH environment variable should include Documaker Workstation program files location. Verify that the PC's PATH= variable includes the program files location; such as C:\FAP\DLL.

The normal Documaker Workstation installation process updates the PATH= variable to include the program directory.

Trouble Shooting

- Check your path statement. You should have C:\FAP\DLL (if that is the installed location) in your path statement. Be sure that it is in your path after a reboot.
- Check your working directory. If you are running from an icon make sure the working directory is correct.
- Check your INI settings. The INI files may be missing and you may have an incorrect directory path statements.
- Check to see if the shipping sample (SAMPCO) works. If not, perhaps you need to go through installation again.

Using the ActiveX Component (RacCo)

Library Platforms

Win 32	OS/2	Win 16	MVS	Unix	OS/400
RACCO					

Library Header Files

RACAPI.H, CRACLIB.H

Overview of Component

The component code is responsible for…

- Determining current window handle that will be passed to RACLib functions. The RACLib functions create a window from the handle passed to them.
- Calling the RACTerminate function when the component destructs and before additional calls to RACLib functions.
- Loading the RACLib DLL (*RACW32.DLL*) dynamically.

Registering the ActiveX Component

You must register the RACCO.*DLL* file in the Windows registry to make the Remote Access Component (RacCo) available. To do so:

- Within the MS-DOS command prompt change directory to Documaker Workstation program directory.
  cd c:\fap\dll
- Type:
  regsvr32 racco.dll

- Windows will display a message that shows the registration was successful.

Component Classes

Name	Description
RacVw	View an archive
RacEdit	Edit a form set
RacImport	Import transactions
RacProxy	Edit form sets

Using Visual Basic

How to use the component from Visual BASIC

1. **Register component.**

   From the command line, you can run the following program.
   Regsvr32 racco.dll

   If you need to unregister the component, type the following.
   Regsvr32 /u racco.dll

2. **Introduce the type library to the Visual BASIC project.**

   Select the *Project* menu.

   Select *References*.

   Click the checkbox beside RacCo *'V.v'* Type Library. (V.v indicates the version number)

3. **Now you can declare an object of the type *RacVw*.**

   See Example below.

   Public RacLib as RacVw

The RACSetWorkingPath API can be used to set the current working directory to the MRL data set you want your application and DAP to work with. A good place to call this function is on the Form Load event, since the MRL should be the current directory before you can work with data in the directory.

Example

The following sample code fragments retrieves an archived form set. The sample uses the SAMPCO master resource library that ships with Documaker Workstation. It creates a RacVw object and uses the SetWorkingDir method to set the working directory then calls ViewByKey to display the archive list selection window in Documaker Workstation.

First, we declared a RacVw object variable in the Declarations area of the form1 source file.

```
Public vbRacVw As RacVw
```

Next, the RacCo object is created and methods are called in sub-routines of the form1 source file. When the form is loaded, the RacVw object is created and the working path is set.

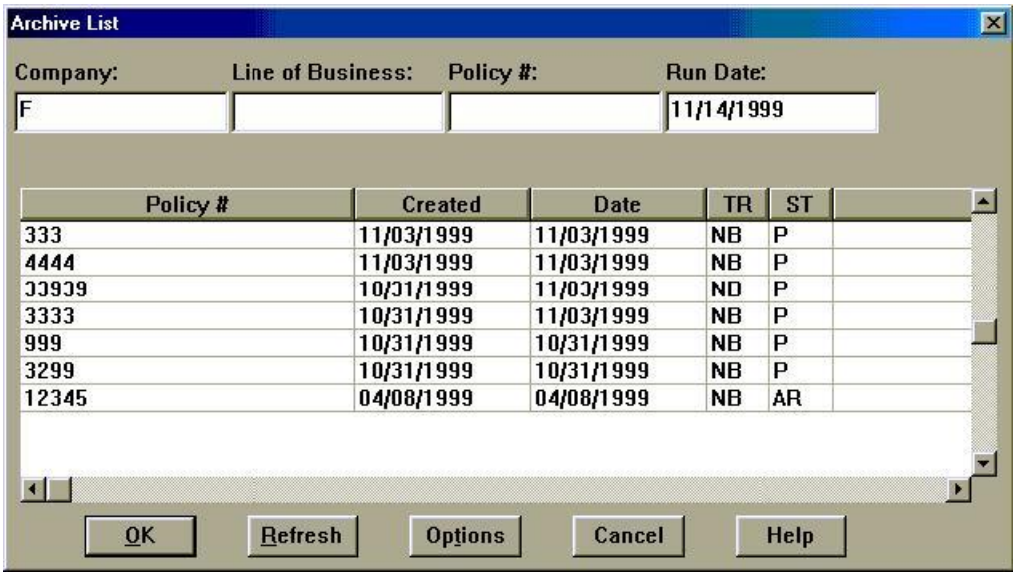
```
Private Sub Form_Load()
```

```
    Set vbRacVw = New RacVw
    vbRacVw.SetWorkingDir "c:\fap\mstrres\sampco"
End Sub
```

When a form button is pressed, RacCo is asked to retrieve archives that have a Key1 value starting with "F".

```
Private Sub cmdRACRetArc_Click()
    vbRacVw.ViewByKey "F", "", "", 1, 0
End Sub
```

When the ViewByKey method executes, a window is created to display the form set in and the following Archive List window appears.



Once a policy is selected, the form set appears in the created window.

Using C++

Files

File Type	Windows
ActiveX DLL	RACCO.DLL
Type Library	RACCO.TLB

Example

The following sample code fragments retrieves an archived form set. The sample uses the SAMPCO master resource library that ships with Documaker Workstation. It creates an IRacVw object and uses the SetWorkingDir method to set the working directory then calls ViewByKey to display the archive list selection window in Documaker Workstation.

First, we create a RacCo object's IRacVw class using the type RACCO*.TLB* library. In the Microsoft Visual Studio you can choose:

- Select the *View* menu.

- Select *Class Wizard…*
- Select *Add Class…*
- Select *From a type library…*
- Locate RACCO*.TLB*

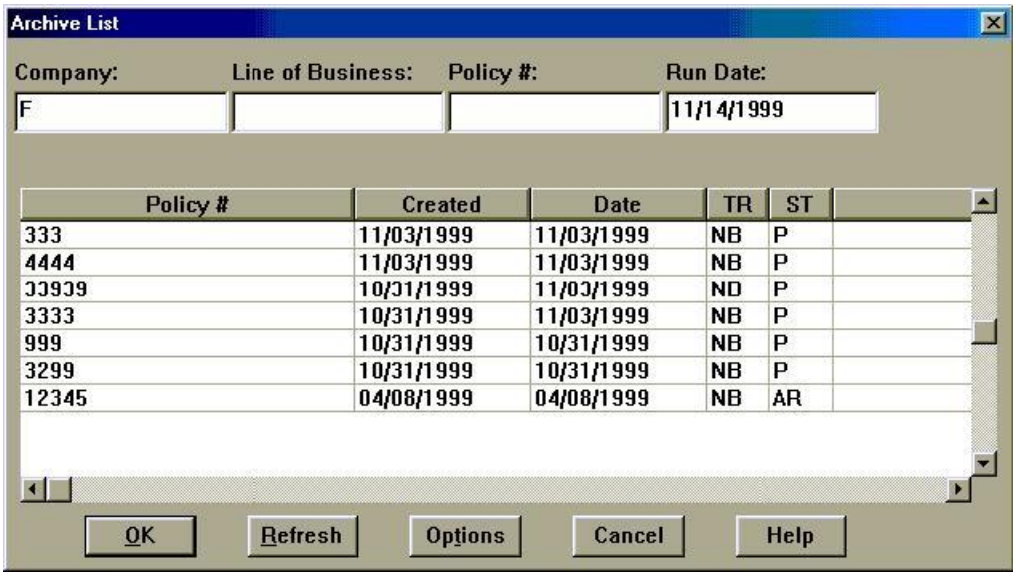
Next, create an instance of the IRacVw object when the application is initialized. Then set the working path for RacCo.

```
pRacVw = new IRacVw();
if (pRacVw != NULL)
{
    pRacVw->CreateDispatch("DocuCorpDAPRacCo.RacVw");
    pRacVw->SetWorkingDir("c:\\fap\\mstrres\\sampco");
}
```

When a menu item is selected, RacCo is asked to retrieve archives that have a Key1 value starting with "F".

```
PRacVw->ViewByKey("F", "", "", 1, 0);
```

When the ViewByKey method executes, a window is created to display the form set in and the following Archive List window appears.



Once a policy is selected, the form set is displayed in the created window.

Using API Functions

Library Platforms

Win 32	MVS	Unix	OS/400
RACW32			

Library Header Files

RACAPI.H, RACLIB.H

Overview of API Functions

Oracle Insurance’s RACLib was created to give non-Oracle Insurance applications the ability to start (and to somewhat control) the form's entry system.

RACLib supports both a modal or non-modal session. A modal session will assume control of window message handling and only return once the user completes form entry. A non-modal session relies upon the controlling application for message handling.

A non-modal session allows an external application to remain responsive to message commands. However, this greater control result in a little more complexity because the application becomes responsible for ensuring that the entry system performs all the necessary operations to collect and save data.

This reference includes those functions exported by RACLib for use by external applications. External applications can either include the *RACAPI.H* header into the source and link with the RAC import library (*RACW32.LIB*), or manually load the DLL thus querying the procedure address by name before each call. The examples in this documentation will assume the application is linked with RACLib.

For those applications that attempt to query procedures by name, it will still be important to note the prototypes of these functions. Calling a function indirectly (as this manner suggests) does not ensure the correct calling convention is used or that the correct size of parameter is pushed. When in doubt, provide a proper variable cast (especially when a "long" variable is required under Windows).

Most functions are prototyped with the calling convention defined by _RACAPI, others use EXPENTRY defines.

The definition of _RACAPI is:

```
#ifdef WIN32
#define _RACAPI WINAPI
#define _RACAPIPTR WINAPI
#endif
```

The type EXPENTRY is defined for Windows as:

```
#ifdef __WIN__
    #define     EXPENTRY    far pascal
#endif
```

Please note that _VMMAPI for Windows 32-bit is defined as nothing.

```
#ifdef WIN32
    #define _VMMAPI
#endif
```

This is included for platform compatibility and may change in the future.

Archive Functions

Function	Description
RACRetrieve	Display transaction from archive.
RACRetrieveArchive	View form set specified.
RACRetrieveArchiveHab	View form set specified.
RACViewByKey	View form set specified.
RACViewByKeyHab	View form set specified.
RACViewData	View form set specified.
RACViewDataHab	View form set specified.

Entry Functions

Function	Description
RACCreate	Create a New WIP Entry
RACCreateEntry	Create a New WIP Entry
RACCreateWipEntry	Create a New WIP Entry
RACEditData	View form set specified.
RACEntry	Start the Entry (PPS) System.
RACMain	Remote Start of Entry (MODAL)
RACSave	Save transaction.
RACThread	New Process operating within another window.
RACUpdate	Edit an existing WIP entry.
RACWipSelectFunction	WIP selection support function.
RACWipSelection	Support function for WIP selection

Error Functions

Function	Description
RACGetDescription	Return description of error message
RACGetStatus	Get Last Error Status Code
RACSetStatus	Set the Error Status Code

Form Set Functions

Function	Description
RACCountPages	Count the pages in a form set
RACFindCompanyLOB	Locate a specified Key1 and Key2 values
RACFindTransaction	Locate a specified transaction

Proxy Functions

The proxy file is the file representation of one DAP transaction. Usually, it is the archive transaction. This file has enough information for the Power Office application to be able to get the key information for display. Additionally, if needed, it has form names, section names, field data, and so on, so the application can invoke some kind of text search mechanism. Therefore, proxy files can be thought of as another import/export file format.

Function	Description
RACArchive2Proxy	Display transaction from archive.
RACSaveAsProxy	Save the current form set in the current AFEData structure into a proxy file.
RACSetProxyHandle	Set the document handle of a proxy file for RACViewProxy.
RACViewProxy	Display archive from proxy file.

Session Functions

Multiple Session RACLib allows Documaker Workstation to have concurrent multiple desktops within the same process. Each window can perform different WIP and ARCHIVE functions within the same process concurrently and independent of the other session. A window's handle identifies each session. The RACInit function attaches Documaker Workstation to a window's handle. This window's handle can be obtained by calling RACGetFrameWindow.

The window's handle returned by RACGetFrameWindow will change after each successive call to RACInit.

RACInit can be called from another function within RACLib or external to RACLib. The following RACLib functions call RACInit internally. Therefore, if you make successive calls to any of these APIs we have new session.

- RACViewData
- RACViewByKey
- RACRetrieveArchive
- RACMain
- RACInitCtrl
- RACInitAll
- RACThread

The maximum number of sessions can be defined with the following INI option:

This example sets the maximum to five sessions. Therefore, the sixth call to RACInit will fail. The current version of RACLib will support no more than 10 concurrent sessions.

Function	Description
RACGetSession	Get the session for this window
RACSetCurSession	Set this window to be the current session.
RACSetSessionMenu	Set the menu's handle in the session structure
RACTermSession	Terminate Entry Session.

System Functions

Command Line Arguments

This module parses command line arguments and keeps the valid arguments in a static structure.

The pointer to the structure is initially NULL but it's filled by the RACSetCmdLineArg function. The memory for the structure is static but there is a linked list's handle that is created via VMMCreateList. This handle will not survive RACTerminate. Therefore, RACSetCmdLineArg will need to be called before each RACInit.

This pointer to the command line structure is set to static memory when RACSetCommandLineArg is called successfully.

Function	Description
RACClose	Close.
RACGetAFEData	Get the Entry Data Structure

RACInit	Initialize Entry System
RACInitAll	Initialize Entire Entry System.
RACInitCtrl	Initialize Entry System (menu and accelerator control).
RACLibVersion	Get library version information.
RACLoadIni	Load the INI file.
RACPackDatabase	Pack the database.
RACRestorePath	Restore original working directory.
RACSetCmdLineArg	This allows historical command line options for AFEMAIN program to be passed to RACInit.
RACSetIniFile	Sets the INI file path name used by RACLib functions.
RACSetWorkingPath	Set current working directory.
RACTerminate	Terminate Entry Session.
RACWorkingPath	Establish correct working directory.

Window Functions

Function	Description
RACCommand	Request command execution.
RACEnableMenu	Change to the Entry Menu.
RACGetAccelHandle	Get the Accelerator Table Handle
RACGetClientWindow	Return client window.
RACGetFrameWindow	Return frame window.
RACGetMenuHandle	Get the Entry Menu Handle.
RACGetOrigMenuHandle	Get the Original Menu Handle.
RACHookProc	Windows accelerator hook procedure
RACLoadMenu	This function will query the INI file for the MEN.RES compatible file to load.
RACMainWndProc	Documaker Workstation message handler.
RACModal	Start Modal Entry.
RACRemoveScrollbar	Remove the scroll bar from window.
RACRestoreMenu	Restore the Original Menu.
RACSetCaptionOff	Set caption off for this window.
RACSetCaptionOn	Set caption on for this window.
RACSetHook	Establish Windows message hook.
RACSetParent	Set parent window.
RACSubClass	Subclass a window's procedure with RACMainWndProc.
RACUnhook	Remove Windows message hook.

Using C/C++

The Remote Access Library (RACLib) was created using the C/C++ computer language. It is a natural fit to call the API functions from your C source. You can include the *RACAPI.H* header file and link the import library. The **Function Reference** in this document details the functions and their declaration syntax.

Files

File Type	Windows
Include file	RACAPI.H
Import library	RACW32.LIB
DLL	RACW32.DLL

Example

The following sample code fragments retrieves an archived form set. The sample uses the SAMPCO master resource library that ships with Documaker Workstation. It sets the working directory by using the RACSetWorkingPath API function then calls RACRetrieveArchive to display the archive list selection window in Documaker Workstation.

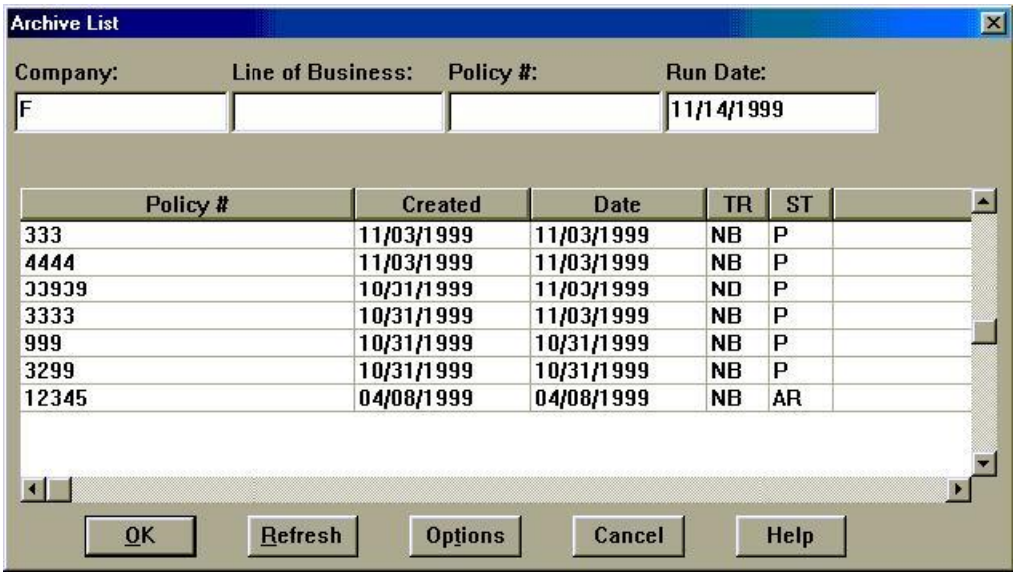
First, the RACLib header file *RACAPI.H* is included.

```
#include "racapi.h"
```

Next, the RACLib API functions are called in window procedure code when a menu item is selected. The working path is set then RACLib is asked to retrieve archives that have a Key1 value starting with "F".

```
RACSetWorkingPath("c:\\fap\\mstrres\\sampco");
RACRetrieveArchive(hWnd,"F","","",1,0);
```

When the RACRetrieveArchive function executes, a window is created to display the form set in and the following Archive List window appears.



Once a policy is selected, the form set is displayed in the created window.

Using Visual Basic

To make direct function calls into RACLib, you must declare the API's to be used in the General Declarations section of the Visual Basic project. The RACLIB.*BAS* file has been provided that declares all the Remote Access API's. You can include or copy the contents of the RACLIB.*BAS* file into your project. The **Function Reference** in this document details the functions and their declaration syntax.

Files

File Type	Windows
Declaration file	RACLIB.BAS
DLL	RACW32.DLL

Example

The following sample code fragments retrieves an archived form set. The sample uses the SAMPCO master resource library that ships with Documaker Workstation. It sets the working directory by using the RACSetWorkingPath API function then calls RACRetrieveArchive to display the archive list selection window in Documaker Workstation.

First, the RACLib API functions are declared in the Declarations area of the form1 source file. These declarations can be found in RACLIB.*BAS.*

```
Private Declare Sub RACSetWorkingPath Lib "racw32.dll" _
                (ByVal filename As String)

Private Declare Function RACRetrieveArchive Lib "racw32.dll" _
                (ByVal hwndParent As Long, _
                 ByVal Key1 As String, _
                 ByVal Key2 As String, _
                 ByVal keyID As String, _
                 ByVal startpage As Long, _
                 ByVal location As Long) As Long
```

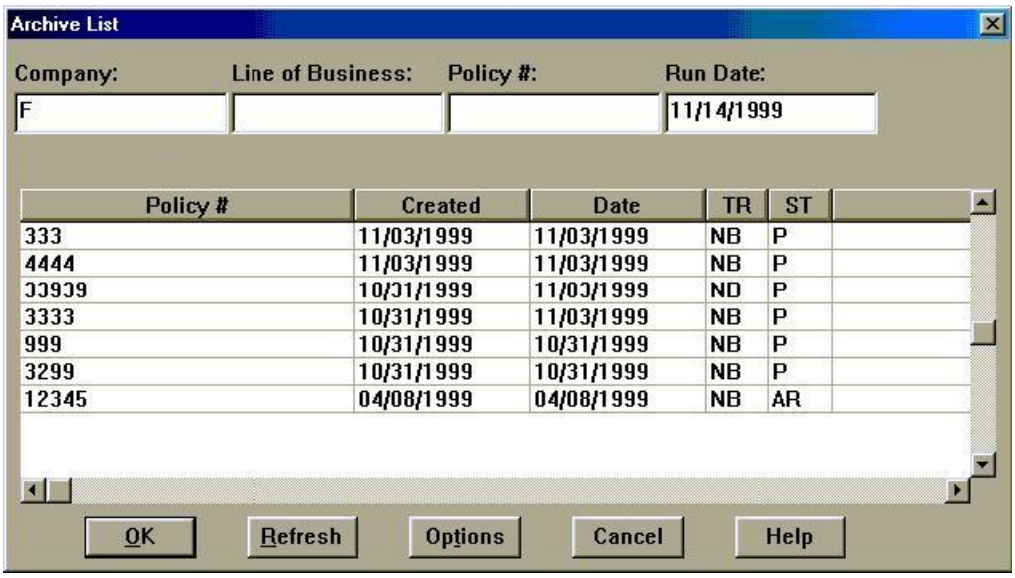
Next, the RACLib API functions are called in sub-routines of the form1 source file. When the form is loaded, the working path is set.

```
Private Sub Form_Load()
    RACSetWorkingPath "c:\fap\mstrres\sampco"
End Sub
```

When a form button is pressed, RACLib is asked to retrieve archives that have a Key1 value starting with "F".

```
Private Sub cmdRACRetArc_Click()
   RACRetrieveArchive Form1.hwnd, "F", "", "", 1, 0
End Sub
```

When the RACRetrieveArchive function executes, a window is created to display the form set in and the following Archive List window appears.



Once a policy is selected, the form set is displayed in the created window.

Using ORACLE Forms

You can make direct function calls into RACLib from ORACLE Forms. You can specify the API's to be used in a *globalvar* package specification and package body. The **Function Reference** in this document details the functions and their declaration syntax.

Example Code

This is how the RACThread function has been accessed from Oracle forms.

The *globalvar* package specification and package body was defined under "PROGRAM UNITS" in a specific form that accesses the RACLib functions. A pushbutton was defined in the Oracle form to start the RACThread function. An Oracle package is a set of functions and data that is grouped together. The package idea is similar to a C++ class.

This is the specification section similar to a class specification defined in the include file.

```
PACKAGE globalvar IS
threadactive NUMBER;
fh_mylib ora_ffi.libHandleType;
fh_windll ora_ffi.libHandleType;
fh_thread ora_ffi.funcHandleType;
fh_terminate ora_ffi.funcHandleType;
fh_getparent ora_ffi.funcHandleType;
END;
```

The *body* section contains function names and code that will be executed when the package is instantiated. In the *globalvar* package, there are no functions but the DLLs are loaded and the functions registered.

```
PACKAGE BODY globalvar IS
BEGIN
threadactive := 0;
fh_mylib := ora_ffi.load_library('E:REL10\FAP400\SHIPW32\','racw32.dll');
fh_windll := ora_ffi.load_library('D:\WINDOWS\SYSTEM\','user32.dll');
fh_thread := ora_ffi.register_function(fh_mylib, 'RACThread', ora_ffi.PASCAL_STD);
```

```
fh_terminate := ora_ffi.register_function(fh_mylib, 'RACTerminate',
ora_ffi.PASCAL_STD);
fh_getparent := ora_ffi.register_function(fh_windll, 'GetParent',
ora_ffi.PASCAL_STD);
ora_ffi.register_parameter(fh_thread, ORA_FFI.C_INT);
ora_ffi.register_return(fh_thread, ORA_FFI.C_INT);
ora_ffi.register_parameter(fh_terminate, ORA_FFI.C_INT);
ora_ffi.register_return(fh_terminate, ORA_FFI.C_INT);
ora_ffi.register_parameter(fh_getparent, ORA_FFI.C_INT);
ora_ffi.register_return(fh_getparent, ORA_FFI.C_INT);
END globalvar;
```

The following code was placed in the trigger section under a WHEN_BUTTON_PRESSED event. We obtain the parent of the current window so the RACThread window will not be placed inside the pushbutton.

```
declare
FUNCTION i_TestDll(funcHandle in ora_ffi.funcHandleType,
aNumber in BINARY_INTEGER) RETURN BINARY_INTEGER;
pragma interface(c, i_TestDll, 11265);
FUNCTION i_Term(funcHandle in ora_ffi.funcHandleType,
aNumber in BINARY_INTEGER) RETURN BINARY_INTEGER;
pragma interface(c, i_Term, 11265);
windowH BINARY_INTEGER;
begin
if globalvar.threadactive = 1 then
windowH := i_Term(globalvar.fh_terminate, windowH);
end if;
windowH := Get_Item_Property(name_in('SYSTEM.CURSOR_ITEM'), WINDOW_HANDLE);
windowH := i_TestDll(globalvar.fh_getparent, windowH);
windowH := i_TestDll(globalvar.fh_thread, windowH);
globalvar.threadactive := 1;
end;
```

Tips and Techniques

Verify that you can access Documaker Workstation before you customize it with RACLib/RacCo.

Run Documaker Workstation

Before beginning to use the Remote Access Library to customize Documaker Workstation, you **must** verify that Documaker Workstation works in your environment. You should be able to run the *AFEMNW32.EXE* (if using Win32 product) to access your archives and do forms entry (if needed in your environment). For more information on setting up Documaker Workstation, refer to the Documaker Workstation Administration Guide and the Documaker Workstation User Guide.

Path statement

When you write your own application to access Documaker Workstation your program will need to locate and load DLLs and INI files that Documaker Workstation uses. Your application will make calls into RACLib, which will then call into other DLL files of Documaker Workstation. Therefore, the PATH environment variable should include Documaker Workstation program files location. Verify that the PC's PATH= variable includes the program files location; such as C:\FAP\DLL.

The normal Documaker Workstation installation process updates the PC's PATH= variable to include the program directory.

Trouble Shooting

- Check your path statement. You should have C:\FAP\DLL (if that is the installed location) in your path statement. Be sure that it is in your path after a reboot.
- Check your working directory. If you are running from an icon make sure the working directory is correct.

- Check your INI settings. The INI files may be missing and you may have incorrect directory path statements.
- Check to see if the shipping sample (SAMPCO) works. If not, perhaps you need to go through installation again.

Check that the working directory and INI file settings are always accessible from your application.

Documaker Workstation uses the working directory path to locate the *FSIUSER.INI* and *FSISYS.INI* files. Within these INI files are path and file settings that the system uses to locate numerous other files and resources. When you are remotely controlling Documaker Workstation, you need to be aware of this working directory system requirement. Therefore, your application should set the working directory path for RACLib/RacCo.

If your application can change its working directory while running, this can cause issues with RACLib/RacCo finding files and resources. For instance, a simple file open window can change the current working directory of an application.

RACLib/RacCo can have trouble finding files and resources if your INI files contain settings with relative paths and your application has changed its working directory while running. If you have this problem, try one of these possible solutions:

- Change your INI files to have absolute path settings
- Save and restore the working path around the process that can change your path. For instance, you could get the current working path before the file open window then restore it once finished.
- Set the working directory for RACLib/RacCo before running each remote access process (*RAC*)

ActiveX Object Reference

AfeProxy (IAfeProxy)

AFE proxy class

Syntax

#include <AFEPROXY.H>
class AfeProxy : public CRacBase

Remarks

The proxy file is the file representation of one DAP transaction. Usually, it is the archive transaction. This file has enough information for the Power Office application to be able to get the key information for display. Additionally, if needed, it has form names, section names, field data, and so on, so the application can invoke some kind of text search mechanism. Therefore, proxy files can be thought of as another import/export file format.

There are two proxy objects in the RacCo component.

- RacProxy

  Queries information from the proxy file, like getting the key information.

- AfeProxy

  Displays an archived transaction based on the input proxy file. Additionally, it lets you create a proxy file from an archived transaction.

Overview

Related Classes

CRacBase

Member Functions

Method	Description
CloseFile	Close proxy file.
GetDocHandle	Get the Documaker Workstation document handle.
GetIndex	Get the index value for a field.
LoadFile	Load a proxy file.
LoadFileVar	Load proxy file.

Details of AfeProxy class

Constructors

public AfeProxy(void)

CloseFile

Close proxy file.

Syntax

public void CloseFile(void)

GetDocHandle

Get the Documaker Workstation document handle.

Syntax

public void GetDocHandle(long* docH)

Parameter	Description
docH	Document handle.

GetIndex

Get the index value for a field.

Syntax

public void GetIndex(BSTR fldName,
BSTR* fldText)

Parameter	Description
fldName	Field name.
fldText	Field text value.

LoadFile

Load a proxy file.

Syntax

public void LoadFile(BSTR fileName)

Parameter	Description
fileName	File name.

LoadFileVar

Load proxy file.

Syntax

public void LoadFileVar(VARIANT* fileName)

Parameter	Description
fileName	File name.

RacEdit (IRacEdit)

Use this function to run Documaker Workstation for form entry.

Syntax

```
#include <RACEDIT.H>
class RacEdit : public CRacBase
```

Overview

Remarks

Use this component to run Documaker Workstation for form entry. The component is an "in-process" DLL implemented with ATL. RACLib is responsible for the actual functionality of the component.

Related Classes

CRacBase

Member Functions

Method	Description
RACMain	Starts a modal session of the form entry system.
RACThread	Starts a new process operating within another window.
SetINIPath	Sets the path that Documaker Workstation uses to locate the INI files.
SetParent	Use this function to set the parent window.
SetRacLibDll	Set path and file name for RACLib.
SetWorkingDir	Set working directory for RACLib DLL.

Details of RacEdit class

Constructors

public RacEdit(void)

RACMain

Starts a modal session of the form entry system.

Syntax

```
public void RACMain(BSTR transaction,
                    BSTR company,
                    BSTR lob,
                    BSTR policy,
                    BSTR description,
                    BSTR userid,
                    BSTR sysid,
                    long mode)
```

Parameter	Description
transaction	A string that represents one of the valid [TRANSACTIONS] abbreviations defined in the INI file. For instance: "NB" for New Business "EN" for Endorsement and so on.
company	A string that represents the first WIP key value (which is also the first component of a line from the FORM.DAT). This parameter is usually referred to as the "Company" value.
lob	A string that represents the second WIP key value (which is also the second component of a line from the FORM.DAT). This parameter is usually referred to as the "Line of Business" value.
policy	A string that represents the WIP key ID value. This parameter is usually referred to as the "ID Number" value.
description	A string that should be assigned as the WIP description value.
userid	A string that corresponds to a valid user ID with access to the Entry module.
sysid	Not currently used.
mode	A valid RACLib action. Currently defined: 1 = AFEACTION_CREATE = Create new WIP. 2 = AFEACTION_UPDATE = Update existing WIP.

Remarks

Starts a modal session that is similar to starting Documaker Workstation (*AFEMNW32.EXE*). No other initialization or termination functions need be called by the user's application.

Since this function starts a modal session, it does not return until the session completes. It will take over the main window handling and replace the existing menu. The new menu will be created from the file specified by the [ALTMENU] setting in the INI file. If omitted, the menu will default to the file specified by the [MENU] setting in the INI file.

The original menu will be restored before returning. During initialization, the login screen can appear if the user ID is not specified by the parameter and is not specified in the INI file.

This method calls RACMain in RACLib.

RACThread

Starts a new process operating within another window.

Syntax

public void RACThread(long hndl)

Parameter	Description
hndl	Parent window to start this application

Remarks

It is assumed that the window handle passed represents the area that must contain the program. A frame window is created inside this area. It then starts the Entry module after the usual initialization.

This method calls the RACThread in RACLib.

SetINIPath

Sets the path that Documaker Workstation uses to locate the INI files.

Syntax

public void SetINIPath(BSTR iniPath)

Parameter	Description
iniPath	Path name for INI files.

Remarks

The system will load FSISYS.INI and FSIUSER.INI from that location. This method calls RACSetIniFile in RACLib.

SetParent

Use this function to set the parent window.

Syntax

public void SetParent(HWND hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

This method calls RACSetParent in RACLib.

SetRacLibDll

Set path and file name for RACLib.

Syntax

public void SetRacLibDll(BSTR racLibPath)

Parameter	Description
racLibPath	Library path name.

Remarks

If this method is not used the component will default to *RACW32.DLL*.

SetWorkingDir

Set working directory for RACLib DLL.

Syntax

public void SetWorkingDir(BSTR workDir)

Parameter	Description
workDir	Working directory path.

Remarks

This method also sets the FSIUSER environment variable. It calls RACSetWorkingPath in RACLib.

RacImport (IRacImport)

Use this function to run Documaker Workstation for form entry.

Syntax

```
#include <RACIMPOR.H>
class RacImport : public CRacBase
```

Overview

Remarks

Use this component to run Documaker Workstation for form entry. The component is an "in-process" DLL implemented with ATL. RACLib is responsible for the actual functionality of the component.

Related Classes

CRacBase

Member Functions

Method	Description
RACEditData	View and edit a form set.
SetINIPath	Sets the path that Documaker Workstation uses to locate the INI files.
SetRacLibDll	Set path and file name for RACLib.
SetWorkingDir	Set working directory for RACLib DLL.

Details of RacImport class

Constructors

```
public RacImport(void)
```

RACEditData

View and edit a form set.

Syntax

```
public void RACEditData(long hndl,
                        BSTR fileName)
```

Parameter	Description
hndl	Parent window to contain this application.
fileName	Name of file to import.

SetINIPath

Sets the path that Documaker Workstation uses to locate the INI files.

Syntax

public void SetINIPath(BSTR iniPath)

Parameter	Description
iniPath	Path name for INI files.

Remarks

The system will load FSISYS.INI and FSIUSER.INI from that location. This method calls RACSetIniFile from RACLib.

SetRacLibDll

Set path and file name for RACLib.

Syntax

public void SetRacLibDll(BSTR racLibPath)

Parameter	Description
racLibPath	Library patch name.

Remarks

If this method is not used the component will default to racw32.dll.

SetWorkingDir

Set working directory for RACLib DLL.

Syntax

public void SetWorkingDir(BSTR workDir)

Parameter	Description
workDir	Working directory path.

Remarks

This method also sets the FSIUSER environment variable.

RacProxy (IRacProxy)

Use this function to run Documaker Workstation for form entry.

Syntax

```
#include <RACPROXY.H>
class RacProxy : public CRacBase
```

Overview

Remarks

Use this component to run Documaker Workstation for form entry. The component is an "in-process" DLL implemented with ATL. RACLib is responsible for the actual functionality of the component.

Related Classes

CRacBase

Member Functions

Method	Description
CloseView	Close the form set
RACArchive2Proxy	Display transaction from archive.
RACCommand	Request command execution.
RACSetCaptionOff	Set caption off for this window.
RACSetCaptionOn	Set caption on for this window.
RACViewAfeProxy	Display archive from proxy file.
RACViewProxy	Display archive from proxy file.
RemoveScrollBar	Remove the scroll bar from window.
SetINIPath	Sets the path that Documaker Workstation uses to locate the INI files.
SetRacLibDll	Set path and file name for RACLib.
SetWorkingDir	Set working directory for RACLib DLL.

Details of RacProxy class

Constructors

public RacProxy(void)

CloseView

Close the form set

Syntax

public void CloseView(void)

Remarks

Close the form set in the RACViewProxy session but don't end the session, by keeping the session alive we can avoid doing another initialization for RACLib.

RACArchive2Proxy

Display transaction from archive.

Syntax

public void RACArchive2Proxy(long hwnd,
BSTR Key1,
BSTR Key2,
BSTR KeyID,
short startpage,
BSTR* proxyfile)

Parameter	Description
hwnd	The parent window to contain this application.
Key1	Search data for Key1.
Key2	Search data for Key2.
KeyID	Search data for KeyID.
startpage	Page to begin display of form set.
proxyfile	Character pointer that contains name of the proxy file

Remarks

Display an archived transaction and allow user to store a proxy file pointing to the transaction in archive. Transactions are filtered the same as RACRetrieveArchive.

This method calls RACArchive2Proxy in RACLib.

RACCommand

Request command execution.

Syntax

public void RACCommand(long cmd)

Parameter	Description

cmd	Command ID

Remarks

This will send the main window a command message corresponding to the command ID passed as a parameter. Most often, this function will be used during a non-modal session to pass in the IDs associated with WM_COMMAND messages. You must initialize the system before you call this function.

This method calls RACCommand in RACLib.

RACSetCaptionOff

Set caption off for this window.

Syntax

public void RACSetCaptionOff(void)

Remarks

Set caption off for this window. This method calls RACSetCaptionOff in RACLib.

RACSetCaptionOn

Set caption on for this window.

Syntax

public void RACSetCaptionOn(void)

Remarks

Set caption on for this window. This method calls RACSetCaptionOn in RACLib.

RACViewAfeProxy

Display archive from proxy file.

Syntax

```
public void RACViewAfeProxy(long hwnd,
                            AfeProxy* obj,
                            short page,
                            short isChildWindow)
```

Parameter	Description
hwnd	Must be a valid window handle to contain form set display.
page	Page to begin display of form set.

Remarks

This method calls RACViewProxy in RACLib.

RACViewProxy

Display archive from proxy file.

Syntax

public void RACViewProxy(long hwnd,
BSTR filename,
short page,
short isChildWindow)

Parameter	Description
hwnd	Must be a valid window handle to contain form set display.
filename	path to a valid proxy file
page	page to begin display of form set

Remarks

This method calls RACViewProxy in RACLib.

RemoveScrollBar

Remove the scroll bar from window.

Syntax

public void RemoveScrollBar(void)

Remarks

Remove the scroll bar from window. This method calls RACRemoveScrollbar in RACLib.

SetINIPath

Sets the path that Documaker Workstation uses to locate the INI files.

Syntax

public void SetINIPath(BSTR iniPath)

Parameter	Description
iniPath	Path name for INI files.

Remarks

The system will load FSISYS.INI and FSIUSER.INI from that location. This method calls RACSetIniFile from RACLib.

SetRacLibDll

Set path and file name for RACLib.

Syntax

public void SetRacLibDll(BSTR racLibPath)

Parameter	Description
racLibPath	Library patch name.

Remarks

If this method is not used the component will default to racw32.dll.

SetWorkingDir

Set working directory for RACLib DLL.

Syntax

public void SetWorkingDir(BSTR workDir)

Parameter	Description
workDir	Working directory path.

Remarks

This method also sets the FSIUSER environment variable.

RacVw (IRacVw)

Use this function to view archived form sets.

Syntax

```
#include <RACVW.H>
class RacVw : public CRacBase
```

Overview

Remarks

Use this component to view archived form sets. The component is an "in-process" DLL implemented with ATL. RACLib is responsible for the actual functionality of the component.

Related Classes

CRacBase

Member Functions

Method	Description
SetINIPath	Sets the path that Documaker Workstation uses to locate the INI files.
SetParent	Use this function to set the parent window.
SetRacLibDll	Set path and file name for RACLib.
SetWorkingDir	Set working directory for RACLib DLL.
ViewByArcKey	Find transactions based on CAR Key.
ViewByFilename	View transaction stored in the export file.
ViewByKey	View the form set that matches the key values supplied.

Details of RacVw class

Constructors

public RacVw(void)

SetINIPath

Sets the path that Documaker Workstation uses to locate the INI files.

Syntax

public void SetINIPath(BSTR iniPath)

Parameter	Description
iniPath	Path name for INI files.

Remarks

The system will load FSISYS.INI and FSIUSER.INI from that location. This method calls RACSetIniFile from RACLib.

SetParent

Use this function to set the parent window.

Syntax

public void SetParent(HWND hwnd)

Remarks

This method calls RACSetParent from RACLib.

SetRacLibDll

Set path and file name for RACLib.

Syntax

public void SetRacLibDll(BSTR racLibPath)

Parameter	Description
racLibPath	Library patch name.

Remarks

If this method is not used the component will default to racw32.dll.

SetWorkingDir

Set working directory for RACLib DLL.

Syntax

public void SetWorkingDir(BSTR workDir)

Parameter	Description
workDir	Working directory path.

Remarks

This method also sets the FSIUSER environment variable.

ViewByArcKey

Find transactions based on CAR Key.

Syntax

```
public void ViewByArcKey(BSTR carkey,
                         short page,
                         short location)
```

Parameter	Description
carkey	Key value of CAR key.
page	Page number.
location	Screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

This method calls RACViewByKey from RACLib.

ViewByFilename

View the transaction stored in the export file.

Syntax

```
public void ViewByFilename(BSTR filename,
                           short page,
                           short location)
```

Parameter	Description
filename	Name of export file.
page	Page number.
location	Screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

This method calls RACViewData from RACLib.

ViewByKey

View the form set that matches the key values supplied.

Syntax

```
public void ViewByKey(BSTR Key1,
                      BSTR Key2,
                      BSTR KeyID,
                      short page,
                      short location)
```

Parameter	Description
Key1	Search data for Key1. Usually known as company.
Key2	Search data for Key2. Usually known as line of business.
KeyID	Search data for KeyID. Usually known as policy number.
page	set beginning page
location	Set screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

Find transactions that match Key1, Key2, and KeyID. The fields that are used for these keys are defined in the INI file. See the *ArcRet* control group options Key1, Key2, and KeyID. Method calls RACRetrieveArchive from RACLib.

Class Reference

AfeLib

Interface class to *AFELib*.

Syntax

```
#include <CAFELIB.H>
class AfeLib
```

Overview

Member Functions

Method	Description
AFECloseProxyFile	Close proxy file.
AFEGetProxyFieldData	Get proxy field data.
AFEIndex2Fld	Get Field given an index
AFEOpenProxyFile	Open proxy file.

Details of AfeLib class

Constructors

public AfeLib(void)

AFECloseProxyFile

Close proxy file.

Syntax

public void AFECloseProxyFile(VMMHANDLE docH)

Parameter	Description
docH	Document handle.

AFEGetProxyFieldData

Get proxy field data.

Syntax

public int AFEGetProxyFieldData(VMMHANDLE docH,
char* fldName,
char* flddata,
char** pAttr,
int fldsize)

Parameter	Description
docH	Document handle.
fldName	Field name.
flddata	Field data.
pAttr	Attributes.
fldsize	Field size.

AFEIndex2Fld

Get Field given an index

Syntax

public void AFEIndex2Fld(enum ProxyIndexFields fldno,
char* fldName,
short fldSize)

Parameter	Description
fldno	Field number.
fldName	Field name.
fldSize	Field size.

AFEOpenProxyFile

Open proxy file.

Syntax

public VMMHANDLE AFEOpenProxyFile(public VMMHANDLE AFEOpenProxyFile@char* fileName)

Parameter	Description
fileName	File name.

CRacBase

Base class for common classes used by RacCo.

Syntax

```
#include <RACBASE.H>
class CRacBase
```

Overview

Remarks

The ActiveX components are derived from this class. It keeps track of the parent window handle and the CRacLib interface class to RACLib.

Related Classes

AfeProxy | RacEdit | RacImport | RacProxy | RacVw

Member Functions

Method	Description
ConvertToUnicode	Convert a string to Unicode.
DestroyProblemWindow	Destroy any problem window that may have been left.
Error	Error function.
GetMyHab	Get the current application instance handle.
GetMyWindow	Get the current active window.
makeWindow	Create a window for Documaker Workstation.
RACCommand	Request command execution.
RACSetCaptionOff	Set caption off for this window.
RACSetCaptionOn	Set caption on for this window.
SetINIPath	Sets the path that Documaker Workstation uses to locate the INI files.
SetParent	Use this function to set the parent window.
SetRacLibDll	Set path and file name for RACLib.
SetWorkingDir	Set working directory for RACLib DLL.

Details of CRacBase class

Constructors

protected CRacBase(void)

Protected Data Members

hwndParent

HWND hwndParent

Remarks

Parent window handle.

RacLib

CRacLib* RacLib

Remarks

Interface to RACLib API functions.

ConvertToUnicode

Convert a string to Unicode.

Syntax

protected OLECHAR* ConvertToUnicode(char* szA)

Parameter	Description
szA	String to convert.

DestroyProblemWindow

Destroy any problem window that may have been left.

Syntax

protected void DestroyProblemWindow(CRacLib* pRacLib,
CRacErr& err)

Parameter	Description
pRacLib	Pointer to a CRacLib class.
err	Reference to error class.

Remarks

Destroy any problem window that may have been left by RACTerminate. This has happened before and while RACTerminate was fixed.

Error

Error function.

Syntax

protected virtual HRESULT Error(LPCSTR err) pure

Remarks

This method is a virtual method.

GetMyHab

Get the current application instance handle.

Syntax

protected HINSTANCE GetMyHab(void)

GetMyWindow

Get the current active window.

Syntax

protected HWND GetMyWindow(void)

Remarks

This function tries to handle the situation if the current window is not available.

makeWindow

Create a window for Documaker Workstation.

Syntax

protected HWND makeWindow(HWND parentWindowH)

Parameter	Description
parentWindowH	Handle of window to use as parent of new window.

RACCommand

Request command execution.

Syntax

protected HRESULT RACCommand(long cmd)

Parameter	Description
cmd	Command ID

Remarks

This sends the main window a command message corresponding to the command ID passed as a parameter. This method calls RACCommand in RACLib.

RACSetCaptionOff

Set the caption off for this window.

Syntax

protected HRESULT RACSetCaptionOff(long hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

Set the caption off for this window. This method calls RACSetCaptionOff in RACLib.

RACSetCaptionOn

Set the caption on for this window.

Syntax

protected HRESULT RACSetCaptionOn(long hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

Set the caption on for this window. This method calls RACSetCaptionOn in RACLib.

SetINIPath

Sets the path that Documaker Workstation uses to locate the INI files.

Syntax

protected void SetINIPath(BSTR iniPath)

Parameter	Description
iniPath	Path name for INI files.

Remarks

The system loads the FSISYS.INI and FSIUSER.INI files from that location. This method calls RACSetIniFile in RACLib.

SetParent

Use this function to set the parent window.

Syntax

protected void SetParent(long hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

This method calls RACSetParent in RACLib.

SetRacLibDll

Set path and file name for RACLib.

Syntax

protected void SetRacLibDll(BSTR racLibPath)

Parameter	Description
racLibPath	Library path name.

Remarks

If this method is not used the component will default to *RACW32.DLL*.

SetWorkingDir

Set working directory for RACLib DLL.

Syntax

protected void SetWorkingDir(BSTR workDir)

Parameter	Description
workDir	Working directory path.

Remarks

This method also sets the FSIUSER environment variable. It calls RACSetWorkingPath in RACLib.

CRacErr

Error message class

Syntax

```
#include <CRACERR.H>
class CRacErr
```

Overview

Remarks

Translates RACLib error messages to text as well any other error number we need to define.

> The CRacLib constructor throws an object of this type if it is unable to load a portion of the DLL.

Member Functions

Method	Description
GetDescription	Get the error description.
GetErrNum	Get the error code number.

Details of CRacErr class

Constructors

Construct Error object with number.

```
public CRacErr(int RacErrNo,
               CRacLib* pRacLib)
```

Parameter	Description
RacErrNo	Error code number
pRacLib	Interface class to RACLib. This constructor calls RACGetDescription in RACLib.

Remarks

This constructor translates RACLib error messages to text as well any other error number we need to define.

Construct Error object with number.

```
public CRacErr(int RacErrNo)
```

Construct Error object with a string.

```
public CRacErr(char* desc)
public CRacErr(void)
```

GetDescription

Get the error description.

Syntax

```
public LPCSTR GetDescription(void)
```

GetErrNum

Get the error code number.

Syntax

```
public int GetErrNum(void)
```

CRacLib

Interface class to RACLib.

Syntax

```
#include <CRACLIB.H>
class CRacLib
```

Overview

Remarks

This class is responsible for making calls into RACLib API functions. All of the ActiveX components have an instance of this class.

Member Functions

Method	Description
RACArchive2Proxy	Display transaction from archive.
RACClose	Close.
RACCommand	Request command execution.
RACEditData	View form set specified.
RACGetDescription	Return description of error message
RACGetMainWindow	Return frame window.
RACGetSession	Get the session for this window
RACInitCtrl	Initialize the Entry system (menu and accelerator control).
RACLoadIni	Load the INI file.
RACLoadMenu	This function will query the INI file for the MEN.RES compatible file to load.
RACMain	Starts a modal session of the Entry system.
RACNeed2Terminate	Gets terminate state.
RACRemoveScrollbar	Remove the scroll bar from window.
RACRestoreMenu	Restore the Original Menu.
RACRetrieveArchive	View form set specified.
RACSetCaptionOff	Set the caption off for this window.
RACSetCaptionOn	Set the caption on for this window.
RACSetCurSession	Set this window to be the current session.
RACSetIniFile	Sets the path that Documaker Workstation uses to locate the INI files.

RACSetProxyHandle	Set the document handle of a proxy file for RACViewProxy.
RACSetWorkingPath	Set working directory for RACLib DLL.
RACTerminate	Terminate an Entry session.
RACTermSession	Terminate an Entry session.
RACThread	Starts a new process operating within another window.
RACViewByKey	View form set specified.
RACViewData	View form set specified.
RACViewProxy	Display archive from proxy file.
SetParent	Use this function to set the parent window.

Details of CRacLib class

Constructors

public CRacLib(void)
public CRacLib(char* dllname)

Public Data Members

isDLLLoaded

BOOL isDLLLoaded

Remarks

Indicates if RACLib DLL was loaded.

RACArchive2Proxy

Display transaction from archive.

Syntax

```
public void RACArchive2Proxy(HWND hwnd,
                             char* Key1,
                             char* Key2,
                             char* KeyID,
                             int startpage,
                             char* proxyfilename)
```

Parameter	Description
hwnd	The parent window to contain this application.
Key1	Search data for Key1.
Key2	Search data for Key2.
KeyID	Search data for KeyID.

startpage	Page to begin display of form set.
proxyfilename	Character pointer that contains name of the proxy file

Remarks

This method calls RACArchive2Proxy in RACLib. Display an archived transaction and allow user to store a proxy file pointing to the transaction in archive. Transactions are filtered the same as RACRetrieveArchive.

RACClose

Close.

Syntax

```
public void RACClose(HWND hwnd,
                     BOOL querysave,
                     BOOL open)
```

Parameter	Description
hwnd	Handle of window.
querysave	Boolean to indicate if a message should ask to save current form set.
open	Boolean to indicate if it should be opened.

Remarks

This method calls RACClose in RACLib.

Performs AFEClose, thus closing the form set and any windows that were associated with that form set. This lets you use the same session again without doing RACInit. You should call this function before you perform a consecutive view function.

RACCommand

Request command execution.

Syntax

```
public void RACCommand(int cmd)
```

Parameter	Description
cmd	Command ID

Remarks

This will send the main window a command message corresponding to the command ID passed as a parameter. Most often, this function will be used during a non-modal session to pass in the IDs associated with WM_COMMAND messages. You must initialize the system before you call this function.

This method calls RACCommand in RACLib.

RACEditData

View form set specified.

Syntax

public void RACEditData(HWND hwndParent,
char* filename)

Parameter	Description
hwndParent	The parent window to contain this application.
filename	The name of a file to import.

Remarks

It is assumed that the window handle passed represents another application's main window. We'll create a frame window to contain the form set view and start the entry system after usual initialization.

This method calls RACEditData in RACLib.

RACGetDescription

Return description of error message

Syntax

public char* RACGetDescription(int ErrNo)

Parameter	Description
ErrNo	Error code.

Remarks

Get the description or message of an error code. This method calls RACGetDescription in RACLib.

RACGetMainWindow

Return frame window.

Syntax

public HWND RACGetMainWindow(void)

Remarks

Return the top level window handle created by RACThread. This method calls RACGetFrameWindow in RACLib.

RACGetSession

Get the session for this window

Syntax

public void* RACGetSession(HWND hwnd)

Parameter	Description
hwnd	Handle of window

Remarks

Retrieve the session that matches on the main frame's window handle. This method calls RACGetSession in RACLib.

RACInitCtrl

Initialize the Entry system (menu and accelerator control).

Syntax

```
public void RACInitCtrl(HWND hwnd,
                        char* userid,
                        char* sysid,
                        BOOL loadMenu,
                        BOOL loadAccel)
```

Parameter	Description
hwnd	The external application's main (highest level) window.
userid	A pointer to a null-terminated string that corresponds to a valid user ID with access to the entry system.
sysid	Future expansion - not currently used.
loadMenu	If true, the menu is loaded.
loadAccel	If true, the accelerator table is loaded.

Remarks

This method calls RACInitCtrl in RACLib. This causes the entry system to initialize. Initialization includes memory management, FAP, WIP, user login, and the help system.

This function will call RACWorkingPath before starting the initialization process. If an error occurs, RACTerminate will be called and the path restored.

If the process initializes correctly and the load menu flag is true; the defined "menu" will be loaded (although not activated). In addition, the default accelerators will be loaded if the load accelerator flag is true.

If the calling application does not call RACModal, then it should eventually call RACTerminate to shutdown correctly.

During initialization, the login screen can appear if the user ID is not specified by a parameter and is not specified in the INI file.

RACLoadIni

Load the INI file.

Syntax

public void RACLoadIni(void)

Remarks

Load the INI file. The file name is defaulted to FSIUSER.INI if RACSetIniFile did not supply it.

This method calls RACLoadIni in RACLib.

RACLoadMenu

This function will query the INI file for the MEN.RES compatible file to load.

Syntax

public void RACLoadMenu(void)

Remarks

This method calls RACLoadMenu in RACLib. This function will query the INI file for the MEN.RES compatible file to load. The following INI options will be queried in the order shown. Once a valid option has been determined the menu is created but not activated. A call to RACEnableMenu is required to activate the menu.

Order of search to determine menu to load:

[AltMenu]

File

If not located then use:

[MENU]

File

If not located the file MEN.RES will be attempted.

If no menu is located, none will be created and an error is returned.

RACRestoreMenu will restore the original menu. RACTerminate automatically calls this function.

RACMain

Starts a modal session of the form entry system.

Syntax

```
public void RACMain(HWND hwnd,
                    char* transaction,
                    char* company,
                    char* lob,
                    char* policy,
                    char* description,
                    char* userid,
                    char* sysid,
                    long mode)
```

Parameter	Description
hwnd	Handle of window.
transaction	A string that represents one of the valid [TRANSACTIONS] abbreviations defined in the INI file. For instance: "NB" for New Business "EN" for Endorsement and so on.
company	A string that represents the first WIP key value (which is also the first component of a line from the FORM.DAT). This parameter is usually referred to as the "Company" value.
lob	A string that represents the second WIP key value (which is also the second component of a line from the FORM.DAT). This parameter is usually referred to as the "Line of Business" value.
policy	A string that represents the WIP key ID value. This parameter is usually referred to as the "ID Number" value.
description	A string that should be assigned as the WIP description value.
userid	A string that corresponds to a valid user ID with access to the Entry module.
sysid	Not currently used.
mode	A valid RACLib action. Currently defined: 1 = AFEACTION_CREATE = Create new WIP. 2 = AFEACTION_UPDATE = Update existing WIP.

Remarks

Starts a modal session that is similar to starting Documaker Workstation (*AFEMNW32.EXE*). No other initialization or termination functions need be called by the user's application.

Since this function starts a modal session, it does not return until the session completes. It will take over the main window handling and replace the existing menu. The new menu will be created from the file specified by the [ALTMENU] setting in the INI file. If omitted, the menu will default to the file specified by the [MENU] setting in the INI file.

The original menu will be restored before returning. During initialization, the login screen can appear if the user ID is not specified by the parameter and is not specified in the INI file.

This method calls RACMain in RACLib.

RACNeed2Terminate

Gets terminate state.

Syntax

public BOOL RACNeed2Terminate(void)

Remarks

If no RACLib termination is needed, TRUE (1) will be returned, otherwise FALSE (0) is returned.

RACRemoveScrollbar

Remove the scroll bar from a window.

Syntax

public void RACRemoveScrollbar(HWND hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

Remove the scroll bar from a window. This method calls RACRemoveScrollbar in RACLib.

RACRestoreMenu

Restore the original menu.

Syntax

public void RACRestoreMenu(void)

Remarks

This method calls RACRestoreMenu in RACLib.

RACRetrieveArchive

View form set specified.

Syntax

public void RACRetrieveArchive(HWND hwndParent,
char* Key1,
char* Key2,
char* KeyID,
int startpage,
int location)

Parameter	Description
hwndParent	The parent window to contain this application.
Key1	Search data for Key1.

Key2	Search data for Key2.
KeyID	Search data for KeyID.
startpage	set beginning page
location	Set screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

This method calls RACRetrieveArchive in RACLib. It is assumed that the window handle passed represents another application's main window. We'll create a frame window to contain the form set view and start the entry system after the usual initialization.

See the ArcRet control group to determine which fields are matched with Key1, Key2, and KeyID.

RACSetCaptionOff

Set caption off for this window.

Syntax

public void RACSetCaptionOff(HWND hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

Set caption off for this window. This method calls RACSetCaptionOff in RACLib.

RACSetCaptionOn

Set caption on for this window.

Syntax

public void RACSetCaptionOn(HWND hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

Set caption on for this window. This method calls RACSetCaptionOn in RACLib.

RACSetCurSession

Set this window to be the current session.

Syntax

public void RACSetCurSession(HWND hwnd)

Parameter	Description
hwnd	Handle of window

Remarks

Set the main frame's window handle (hwndMainFrame variable) if we have a valid session for the handle passed. This method calls RACSetCurSession in RACLib.

RACSetIniFile

Sets the path that Documaker Workstation uses to locate the INI files.

Syntax

public void RACSetIniFile(char* iniPath)

Parameter	Description
iniPath	Path name for INI files.

Remarks

The system loads the FSISYS.INI and FSIUSER.INI files from that location. This method calls RACSetIniFile in RACLib.

RACSetProxyHandle

Set the document handle of a proxy file for RACViewProxy.

Syntax

public void RACSetProxyHandle(VMMHANDLE docH)

Parameter	Description
docH	Document handle.

Remarks

Set the document handle of a proxy file for RACViewProxy.

This method calls RACSetProxyHandle in RACLib.

RACSetWorkingPath

Set working directory for RACLib DLL.

Syntax

public void RACSetWorkingPath(char* workDir)

Parameter	Description
workDir	Working directory path.

Remarks

This method also sets the FSIUSER environment variable. It calls RACSetWorkingPath in RACLib.

RACTerminate

Terminate Entry Session.

Syntax

public void RACTerminate(void)

Remarks

This method calls RACTerminate in RACLib. This will cause the session to release all resources used by the entry system. You must initialize the system before you call this function.

A non-modal session relies upon the controlling application to request termination. Exiting without terminating the system can cause unpredictable results.

RACTermSession

Terminate Entry Session.

Syntax

public void RACTermSession(BOOL mode,
HWND hwnd)

Parameter	Description
mode	The parameter mode should be TRUE if the entry system should ask the user to save (if necessary) and FALSE to cause the system to shutdown without asking.
hwnd	Handle of window

Remarks

This method calls RACTermSession in RACLib.

This will cause the session to release all resources used by the entry system. You must initialize the system before you call this function.

A non-modal session relies upon the controlling application to request termination. Exiting without terminating the system can cause unpredictable results.

RACThread

Starts a new process operating within another window.

Syntax

public void RACThread(HWND hndl)

Parameter	Description
hndl	Parent window to start this application

Remarks

It is assumed that the window handle passed represents the area that must contain the program. A frame window will be created inside this area. Then starts the Entry module after the usual initialization.

This method calls the RACThread in RACLib.

RACViewByKey

View form set specified.

Syntax

public void RACViewByKey(HWND hwndParent,
char* key,
int startpage,
int location)

Parameter	Description
HwndParent	The parent window to contain this application.
Key	The archives key for the transaction.
Startpage	The page number to start display (defaults to 1)
Location	Screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

This method calls RACViewByKey in RACLib. It is assumed that the window handle passed represents another application's main window. We'll create a frame window to contain the form set view and start the entry system after usual initialization.

RACViewData

View form set specified.

Syntax

```
public void RACViewData(HWND hwndParent,
                        char* filename,
                        int startpage,
                        int location)
```

Parameter	Description
hwndParent	The parent window to contain this application.
filename	The name of a file to import.
startpage	The page number to start display (defaults to 1)
location	Screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

This method calls RACViewData in RACLib. It is assumed that the window handle passed represents another application's main window. We'll create a frame window to contain the form set view and start the entry system after usual initialization.

RACViewProxy

Display archive from proxy file.

Syntax

```
public void RACViewProxy(HWND hwnd,
                         char* filename,
                         int startpage)
```

Parameter	Description
hwnd	must be a valid window handle to contain form set display
filename	path to a valid proxy file
startpage	page to begin display of form set

Remarks

This method calls RACViewProxy in RACLib.

SetParent

Use this function to set the parent window.

Syntax

public void SetParent(HWND hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

This method calls RACSetParent in RACLib.

Function Reference

RACArchive2Proxy

Display transaction from archive.

Syntax

int RACArchive2Proxy(HWND hwnd,
char* Key1,
char* Key2,
char* KeyID,
int startpage,
char* proxyfilename)

Parameter	Description
hwnd	The parent window to contain this application.
Key1	Search data for Key1.
Key2	Search data for Key2.
KeyID	Search data for KeyID.
startpage	Page to begin display of form set.
proxyfilename	Character pointer that contains name of the proxy file

Remarks

Display an archived transaction and allow user to store a proxy file pointing to the transaction in archive. Transactions are filtered the same as RACRetrieveArchive.

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
int _RACAPI RACArchive2Proxy(HWND hwnd,
                        char *Key1,
                        char *Key2,
                        char *KeyID,
                        int startpage,
                        char *proxyfilename);
```

Visual Basic Declaration

```
Private Declare Function RACArchive2Proxy Lib "racw32.dll" _
                (ByVal hwnd As Long, _
                 ByVal Key1 As String, _
                 ByVal Key2 As String, _
                 ByVal KeyID As String, _
```

```
                        ByVal startPage As Long, _
                        ByVal proxyfilename As String) As Long
```

typedef

```
typedef int (_RACAPIPTR RACARCHIVE2PROXY)
                            (HWND hwnd,
                            char *Key1,
                            char *Key2,
                            char *KeyID,
                            int startpage,
                            char *proxyfile);
```

See Also

RACSaveAsProxy, RACViewProxy

Calls

RACGetSession, RACGetStatus, RACModalEx, RACProcessError, RACRetrieve, RACSetCurSession, RACSetStatus

RACClose

Close.

Syntax

int RACClose(HWND hwnd,
unsigned int querysave,
unsigned int open)

Parameter	Description
hwnd	Handle of window.
querysave	Boolean to indicate if a message should ask to save current form set.
open	Boolean to indicate if it should be opened.

Remarks

Performs AFEClose, thus closing the form set and any windows that were associated with that form set. This lets you use the same session again without doing RACInit. You should call this function before performing a consecutive view function.

Files

Include: racapi.h
Source: RACTERM.C

Code Declarations

```
int _RACAPI RACClose(HWND hwnd, BOOL querysave, BOOL open);
```

Visual Basic Declaration

```
Private Declare Function RACClose Lib "racw32.dll" _
                (ByVal hwnd As Long, _
                 ByVal querysave As Long, _
                 ByVal openflag As String) As Long
```

typedef

```
typedef void (_RACAPIPTR RACCLOSE)(HWND hwnd,
                                   BOOL queryclose,
                                   BOOL open);
```

Calls

RACGetSession

RACCommand

Request command execution.

Syntax

int RACCommand(int comID)

Parameter	Description
comID	Command ID

Remarks

This will send the main window a command message corresponding to the command ID passed as a parameter. Most often, this function will be used during a non-modal session to pass in the IDs associated with WM_COMMAND messages. You must initialize the system before you call this function.

Returns

The return value will be that returned from a WndProc in the given operating environment.

Files

Include: racapi.h
Source: RACPROC.C

Code Declarations

```
int _RACAPI RACCommand(int comID);
```

Visual Basic Declaration

```
Private Declare Function RACCommand Lib "racw32.dll" _
                (ByVal comID As Long) As Long
```

typedef

```
typedef int (_RACAPIPTR RACCOMMAND)(int comID);
```

Example

```
The following code is an excerpt from an external program.

    switch ( msg ) {
    case WM_COMMAND:
            rval =  RACCommand(LOWORD(wParam));
            .....
```

Calls

RACCountPages

RACCountPages

Count form set pages

Syntax

int RACCountPages(void)

Remarks

This will return the number of pages in the currently open form set.

Returns

This returns the number of pages. A zero return value indicates that the form set is empty or invalid.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
    int _RACAPI RACCountPages(void);
```

Visual Basic Declaration

```
    Private Declare Function RACCountPages Lib "racw32.dll" () As Long
```

RACCreate

Create a New WIP Entry

Syntax

```
int RACCreate(char* transaction,
        char* company,
        char* lob,
        char* policy,
        char* description)
```

Parameter	Description
transaction	A null-terminated string pointer that represents one of the valid [TRANSACTIONS] abbreviations defined in the INI file. For instance: "NB" for New Business; "EN" for Endorsement; and so on.
company	A null-terminated string pointer that corresponds to the first WIP key value (which is also the first component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Company" value.
lob	A null-terminated string pointer that corresponds to the second WIP key value (which is also the second component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Line of Business" value.
policy	A null-terminated string pointer that corresponds to the WIP KeyID value. This parameter is sometimes referred to as the "ID Number" value.
description	A pointer to a null-terminated string that should be assigned as the WIP description value.

Remarks

This will attempt to create a new WIP entry during a non-modal session by calling RACCreateEntry. You must initialize the system before you call this function.

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACCREAT.C

Code Declarations

```
int  _RACAPI RACCreate( char far *transaction,
                        char far *company,
                        char far *lob,
                        char far *policy,
                        char far *description);
```

Visual Basic Declaration

```
Private Declare Function RACCreate Lib "racw32.dll" _
```

```
                    (ByVal trans As String, _
                     ByVal Key1 As String, _
                     ByVal Key2 As String, _
                     ByVal KeyID As String, _
                     ByVal description As String) As Long
```

Example

```
The following code is an excerpt from an external program.
switch ( msg ) {
case WM_COMMAND:
    switch(Param1) {
    case ID_CREATE_NEWPOL:
        rval = RACCreate("NB",
                         "ACME INSURANCE",
                         "HEALTH",
                         "1402001",
                         "New Sample Policy");
        if (rval != RAC_SUCCESS){
            ...  // handle error
            break;
        }
        ... // continue success
        break;
    }
```

See Also

RACCreateEntry, RACInit

Calls

RACCreateEntry

RACCreateEntry

Create a New WIP Entry

Syntax

```
int RACCreateEntry(char* transaction,
                   char* company,
                   char* lob,
                   char* policy,
                   char* description,
                   unsigned int RunEntry)
```

Parameter	Description
transaction	A null-terminated string pointer that represents one of the valid [TRANSACTIONS] abbreviations defined in the INI file. For instance: "NB" for New Business; "EN" for Endorsement; and so on.
company	A null-terminated string pointer that corresponds to the first WIP key value (which is also the first component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Company" value.
lob	A null-terminated string pointer that corresponds to the second WIP key value (which is also the second component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Line of Business" value.
policy	A null-terminated string pointer that corresponds to the WIP KeyID value. This parameter is sometimes referred to as the "ID Number" value.
description	A null-terminated string pointer that should be assigned as the WIP description value.
RunEntry	A Boolean that specifies whether Entry should begin or not. If Entry is started, the desktop and form windows are created. Whether Entry is started or not, the current form set will be defined in the pdata structure.

Remarks

Create a New WIP Entry. This will attempt to create a new WIP entry during a non-modal session. You must initialize the system before you call this function. RACLib should have been initialized before calling this function.

For additional information see: RACInitAll, RACModal

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACCREAT.C

Code Declarations

```
int _RACAPI RACCreateEntry(char far *transaction,
                           char far *company,
                           char far *lob,
                           char far *policy,
                           char far *description,
                           BOOL RunEntry);
```

Visual Basic Declaration

```
Private Declare Function RACCreateEntry Lib "racw32.dll" _
                (ByVal trans As String, _
                 ByVal Key1 As String, _
                 ByVal Key2 As String, _
                 ByVal KeyID As String, _
                 ByVal description As String, _
                 ByVal runEntry As Long) As Long
```

Example

```
The following code is an excerpt from an external program.
switch ( msg ) {
case WM_COMMAND:
    switch(Param1){
    case ID_CREATE_NEWPOL:
        rval = RACCreateEntry("NB",
                              "ACME INSURANCE",
                              "HEALTH",
                              "1402001",
                              "New Sample Policy",
                              TRUE);
        if (rval != RAC_SUCCESS){
                ...  handle error
               break;
        }
        ... continue success
        break;
    }
```

See Also

RACInitAll, RACModal

Calls

AFELoadPPSFormset, RACCreateWipEntry, RACFindTransaction

RACCreateWipEntry

Create a New WIP Entry

Syntax

int RACCreateWipEntry(char* company,
char* lob,
char* policy,
char* description,
PAFEDATA pdata,
unsigned int uniqcheck)

Parameter	Description
company	A null-terminated string pointer that corresponds to the first WIP key value (which is also the first component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Company" value.
lob	A null-terminated string pointer that corresponds to the second WIP key value (which is also the second component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Line of Business" value.
policy	A null-terminated string pointer that corresponds to the WIP KeyID value. This parameter is sometimes referred to as the "ID Number" value.
description	A null-terminated string pointer that should be assigned as the WIP description value.

Remarks

Create a New WIP Entry.

Files

Include: racapi.h
Source: RACCREAT.C

Code Declarations

```
int _RACAPI RACCreateWipEntry(char far *company,
                              char far *lob,
                              char far *policy,
                              char far *description,
                              PAFEDATA pdata,
                              BOOL     uniqcheck);
```

Visual Basic Declaration

```
Private Declare Function RACCreateWipEntry Lib "racw32.dll" _
                (ByVal Key1 As String, _
                 ByVal Key2 As String, _
                 ByVal KeyID As String, _
                 ByVal description As String, _
                 ByVal pdata As Long, _
                 ByVal uniqcheck As Long) As Long
```

Calls

RACCheckUserEntry

RACEditData

View form set specified.

Syntax

HWND RACEditData(HWND hwndParent,
char* filename)

Parameter	Description
hwndParent	The parent window to contain this application.
filename	The name of a file to import.

Remarks

It is assumed that the window's handle passed represents another application's main window. We'll create a frame window to contain the form set view and start the entry system after usual initialization.

Returns

If successful, the handle of our desktop window is returned. On failure, zero will be returned and the caller should use RACGetStatus to retrieve the last know error code. Error code values are defined in RACAPI.H.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
HWND  _RACAPI RACEditData(HWND hwndParent,
                          char *filename);
```

Visual Basic Declaration

```
Private Declare Function RACEditData Lib "racw32.dll" _
                (ByVal hwndParent As Long, _
                 ByVal filename As String) As Long
```

typedef

```
typedef HWND (_RACAPIPTR RACEDITDATA)
                         (HWND hwndParent, char *filename);
```

See Also

RACCreateMainWindow, RACInitAll

Calls

RACEditDataStat

RACEnableMenu

Change to the Entry Menu.

Syntax

int RACEnableMenu(void)

Remarks

This will cause the session to switch the menu within the main window. You must initialize the system before you call this function. This function replaces the existing menu with a new menu created from the file loaded by RACLoadMenu.

Calling RACRestoreMenu will restore the original menu.

Returns

This returns RAC_SUCCESS upon success or one of the values defined in RACAPI.H.

Files

Include: racapi.h
Source: RACMENU.C

Code Declarations

```
int _RACAPI RACEnableMenu(void);
```

Visual Basic Declaration

```
Private Declare Function RACEnableMenu Lib "racw32.dll" As Long
```

RACEntry

Start the Entry (PPS) System.

Syntax

int RACEntry(HWND hwnd)

Parameter	Description
hwnd	The external application's main (highest level) window.

Remarks

This is the simplest method of causing the Entry Program start. The only parameter required is the handle to a main window that contains (or can contain) a menu. This entry point starts the application in a modal fashion and will return to the caller's application when the user "Exits" from our menu.

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACENTRY.C

Code Declarations

```
    int _RACAPI RACEntry(HWND hwnd);
Visual Basic Declaration
    Private Declare Function RACEntry Lib "racw32.dll" _
                        (ByVal hwnd As Long) As Long
```

Example

```
The following code is an excerpt from an external C program.
switch ( msg ) {
case WM_COMMAND:
        switch(Param1){
        case ID_INIT_ENTRY:
                rval = RACEntry( hwnd );
                if (rval != RAC_SUCCESS){
                        ... handle errors
                        break;
                }
                ... continue success
                break;
        }
In Visual Basic the following code could be used.
Sub Action_Click (Index As Integer)
        retval% = RACEntry(Form1.hWnd)
REM Handle errors (retval% != 0) if needed.
End Sub
```

See Also

RACInitAll, RACTerminate, RACModal

Calls

RACGetStatus, RACInitAll, RACModal, RACProcessError

RACFindCompanyLOB

Locate a Specified Key1 and Key2 values

Syntax

VMMHANDLE RACFindCompanyLOB(VMMHANDLE masterH,
char* company,
char* lob)

Parameter	Description
masterH	Form set handle that will be searched.
company	first group name (a.k.a. Key1)
lob	second group name (a.k.a. Key2)

Remarks

Determines if the specified company and line of business (Key1 and Key2) are a valid form set combination. The search is case insensitive.

> This function is somewhat antiquated and limited in functionality. It has been left intact for legacy systems.

Returns

This returns a VMMHANDLE to the located group or VMMNULLHANDLE if not located.

Files

Include: racapi.h
Source: RACFIND.C

Code Declarations

```
VMMHANDLE _RACAPI RACFindCompanyLOB(VMMHANDLE masterH,
                                    char far *company,
                                    char far *lob);
```

Visual Basic Declaration

```
Private Declare Function RACFindCompanyLOB Lib "racw32.dll" _
                (ByVal masterH As Long, _
                 ByVal company As String, _
                 ByVal lob As String) As Long
```

RACFindTransaction

Locate a Specified Transaction

Syntax

VMMHANDLE RACFindTransaction(VMMHANDLE translistH,
char* transaction)

Parameter	Description
translistH	VMM list of valid transaction structures.
transaction	Transaction being searched.

Remarks

This searches the specified VMM list for the specified transaction. This search is case insensitive.

Returns

A VMMHANDLE to the located transaction or VMMNULLHANDLE if not located is returned.

Files

Include: racapi.h
Source: RACFIND.C

Code Declarations

```
VMMHANDLE _RACAPI RACFindTransaction(VMMHANDLE translistH,
                                     char far *transaction);
```

Visual Basic Declaration

```
Private Declare Function RACFindTransaction Lib "racw32.dll" _
                (ByVal translistH As Long, _
                 ByVal transaction As String) As Long
```

RACGetAccelHandle

Get the Accelerator Table Handle

Syntax

HWND RACGetAccelHandle(void)

Remarks

This returns the accelerator tables handle if it was loaded by a function (directly or indirectly) within RACLib. This is provided for applications that manage non-modal sessions that wish to provide accelerator translations.

Returns

A HANDLE or NULL is returned.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
HWND _RACAPI RACGetAccelHandle(void);
```

Visual Basic Declaration

```
Private Declare Function RACGetAccelHandle Lib "racw32.dll" As Long
```

typedef

```
typedef HWND  ( _RACAPIPTR RACGETACCELHANDLE) (void);
```

RACGetAFEData

Get the Entry Data Structure

Syntax

PAFEDATA RACGetAFEData(void)

Remarks

This returns the AFEDATA structure created by RACInit.

> This structure should not be altered by the calling application. This structure contains most of the system wide information used by the Entry system. This structure is defined by AFELIB.H.

Returns

A pointer to the current AFEDATA structure will be returned.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
PAFEDATA _RACAPI RACGetAFEData(void);
```

Visual Basic Declaration

```
Private Declare Function RACGetAFEData Lib "racw32.dll" As Long
```

RACGetClientWindow

Return client window.

Syntax

HWND RACGetClientWindow(void)

Remarks

Get the top client window's handle created by RACThread.

Returns

The window's handle or NULLHANDLE is returned.

Files

Include: racapi.h
Source: RACTHRD.C

Code Declarations

```
    HWND _RACAPI RACGetClientWindow(void);
```

Visual Basic Declaration

```
    Private Declare Function RACGetClientWindow Lib "racw32.dll" As Long
```

typedef

```
typedef HWND (_RACAPIPTR RACGETCLIENTWINDOW) (void);
```

RACGetDescription

Return description of error message

Syntax

char* RACGetDescription(int RacErrNo)

Parameter	Description
RacErrNo	Error code.

Remarks

Get the description or message of an error code.

Returns

This returns a character pointer to the message.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
    char* _RACAPI RACGetDescription(int RacErrNo);
```

Visual Basic Declaration

```
    Private Declare Function RACGetDescription Lib "racw32.dll" _
                    (ByVal errorCode As Long) As String
```

typedef

```
typedef char* (_RACAPIPTR RACGETDESCRIPTION) (int ErrNo);
```

RACGetFrameWindow

Return frame window.

Syntax

HWND RACGetFrameWindow(void)

Remarks

Return the top level Window Handle created by RACThread.

Files

Include: racapi.h
Source: RACTHRD.C

Code Declarations

```
HWND _RACAPI RACGetFrameWindow(void);
```

Visual Basic Declaration

```
Private Declare Function RACGetFrameWindow Lib "racw32.dll" As Long
```

RACGetMenuHandle

Get the Entry Menu Handle.

Syntax

HWND RACGetMenuHandle(void)

Remarks

This returns the menu's handle of a menu created by RACLoadMenu. You must initialize the system before you call this function.

Returns

A handle to a menu will be returned or NULL.

Files

Include: racapi.h
Source: RACMENU.C

Code Declarations

```
    HWND _RACAPI RACGetMenuHandle(void);
Visual Basic Declaration
    Private Declare Function RACGetMenuHandle Lib "racw32.dll" As Long
```

RACGetOrigMenuHandle

Get the Original Menu Handle.

Syntax

HWND RACGetOrigMenuHandle(void)

Remarks

This returns the original main window's menu handle if it was replaced by a call to RACEnableMenu. You must initialize the system before you call this function.

Returns

A handle to a menu will be returned or NULL. A NULL value means the main application window does not contain a menu or that the original menu has not been replaced by RACLib.

Files

Include: racapi.h
Source: RACMENU.C

Code Declarations

```
    HWND _RACAPI RACGetOrigMenuHandle(void);
Visual Basic Declaration
    Private Declare Function RACGetOrigMenuHandle Lib "racw32.dll" As Long
```

RACGetSession

Get the session for this window

Syntax

struct racdata* RACGetSession(HWND hwnd)

Parameter	Description
hwnd	Handle of window

Remarks

Retrieve the session that matches on the main frame window's handle.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
    struct racdata* _RACAPI RACGetSession(HWND hwnd);
```

Visual Basic Declaration

```
    Private Declare Function RACGetSession Lib "racw32.dll" _
                (ByVal hwnd As Long) As Long
```

typedef

```
typedef void* (_RACAPIPTR RACGETSESSION) (HWND hwnd);
```

RACGetStatus

Get Last Error Status Code

Syntax

int RACGetStatus(void)

Remarks

This returns the most recent error code assigned by RACLib functionality.

Returns

One of the values defined in RACAPI.H is normally returned.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
    int  _RACAPI RACGetStatus(void);
```

Visual Basic Declaration

```
    Private Declare Function RACGetStatus Lib "racw32.dll" As Long
```

typedef

```
typedef int (_RACAPIPTR RACGETSTATUS) (void);
```

RACHookProc

Windows accelerator hook procedure

Syntax

LRESULT RACHookProc(int code,
WPARAM wParam,
LPARAM lParam)

Parameter	Description
code	a Window's hook code value
wParam	the first message parameter
lParam	the second message parameter

Remarks

This function is only useful for Windows applications. RACHookProc determines if the next queued message is an accelerator message for a RACLib application window or dialog.

The message hook is established by a call to RACSetHook and removed via RACUnhook.

Returns

Hooks are called in sequence; therefore, this hook will call the one installed before this hook. The return value will be the result of the last hook that executes. See Windows programming help for the meaning of the return values.

Files

Include: racapi.h
Source: RACHOOK.C

Code Declarations

```
LRESULT CALLBACK RACHookProc(int code,
                             MPARAM1 wParam,
                             MPARAM2 lParam);
```

Visual Basic Declaration

```
Private Declare Function RACHookProc Lib "racw32.dll" _
            (ByVal codeval As Long, _
             ByVal wParam As Long, _
             ByVal lParam As Long) As Long
```

RACInit

Initialize Entry System

Syntax

```
int RACInit(HWND hwnd,
        char* userid,
        char* sysid)
```

Parameter	Description
hwnd	The external application's main (highest level) window.
userid	A null-terminated string pointer that corresponds to a valid user ID with access to the entry system.
sysid	Future expansion - not currently used.

Remarks

This causes the entry system to initialize. Initialization includes memory management, FAP, WIP, user login, and the help system.

Calling application will need to call RACTerminate to shutdown correctly at program end or during failure conditions.

When establishing a non-modal session, only RACWorkingPath can be called before this function.

During initialization, the login screen can appear if the user ID is not specified by the parameter and is not specified in the INI file.

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACINIT.C

Code Declarations

```
int  _RACAPI RACInit(HWND mainWnd,
                     char far *userid,
                     char far *sysid);
```

Visual Basic Declaration

```
Private Declare Function RACInit Lib "racw32.dll" _
            (ByVal mainWnd As Long, _
             ByVal userid As String, _
             ByVal sysid As String) As Long
```

Example

```
The following code is an excerpt from an external program.
```

```
switch ( msg ) {
case WM_COMMAND:
    switch(Param1){
    case ID_INIT_ENTRY:
        rval = RACInit(hwnd, "USER1", "");
        if (rval != RAC_SUCCESS){
                RACTerminate(FALSE);
                break;
        }
        ... // continue success
        break;
    }
    break;
case WM_CLOSE:
    ... // if RAC running
    RACTerminate(FALSE);
    break;
```

See Also

RACTerminate

Calls

RACCmdLineArg2Afe, RACCreateNewSession, RACGetCmdLineArg, RACGetSession, RACSetCurSession, RACSetIniFile, RACSetStatus

RACInitAll

Initialize Entire Entry System.

Syntax

int RACInitAll(HWND hwnd,
char* userid,
char* sysid)

Parameter	Description
hwnd	The external application's main (highest level) window.
userid	A null-terminated string pointer that corresponds to a valid user ID with access to the entry system.
sysid	Future expansion - not currently used.

Remarks

This causes the entry system to initialize. Initialization includes memory management, FAP, WIP, user login, and the help system.

This function will call RACWorkingPath before starting the initialization process. If an error occurs, RACTerminate will be called and the path restored.

If the process initializes correctly, the defined "menu" will be loaded (although not activated) and the default accelerators will be loaded.

If the calling application does not call RACModal, then it should eventually call RACTerminate to shutdown correctly.

During initialization, the login screen can appear if the user ID is not specified by parameter and is not specified in the INI file.

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACINIT.C

Code Declarations

```
int  _RACAPI RACInitAll(HWND mainWnd,
                        char far *userid,
                        char far *sysid);
```

Visual Basic Declaration

```
Private Declare Function RACInitAll Lib "racw32.dll" _
            (ByVal mainWnd As Long, _
             ByVal userid As String, _
             ByVal sysid As String) As Long
```

Example

```
The following code is an excerpt from an external program.
switch ( msg ) {
case WM_COMMAND:
        switch(Param1){
        case ID_INIT_ENTRY:
                rval = RACInitAll(hwnd,
                                        "USER1",
                                        "");
                if (rval != RAC_SUCCESS){
                        ... handle error
                        break;
                }
                ... continue success
                break;
        }
case WM_CLOSE:
   ... if RAC running
   RACTerminate(FALSE);
   break;
```

See Also

RACInit, RACWorkingPath, RACLoadMenu, RACLoadAccelerators, RACTerminate

Calls

RACGetStatus, RACInit, RACLoadAccelerators, RACLoadMenu, RACProcessError, RACRestorePath, RACSetStatus, RACWorkingPath

RACInitCtrl

Initialize Entry System (menu and accelerator control).

Syntax

```
int RACInitCtrl(HWND hwnd,
                char* userid,
                char* sysid,
                unsigned int loadMenu,
                unsigned int loadAccel)
```

Parameter	Description
hwnd	The external application's main (highest level) window.
userid	A null-terminated string pointer that corresponds to a valid user ID with access to the entry system.
sysid	Future expansion - not currently used.
loadMenu	If true, the menu is loaded.
loadAccel	If true, the accelerator table is loaded.

Remarks

This causes the entry system to initialize. Initialization includes memory management, FAP, WIP, user login, and the help system.

This function will call RACWorkingPath before starting the initialization process. If an error occurs, RACTerminate will be called and the path restored.

If the process initializes correctly and the load menu flag is true; the defined "menu" will be loaded (although not activated). In addition, the default accelerators will be loaded if the load accelerator flag is true.

If the calling application does not call RACModal, then it should eventually call RACTerminate to shutdown correctly.

During initialization, the login screen can appear if the user ID is not specified by parameter and is not specified in the INI file.

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACINIT.C

Code Declarations

```
int _RACAPI RACInitCtrl(HWND hwnd,
                        char far *userid,
```

```
                              char far *sysid,
                              BOOL loadMenu,
                              BOOL loadAccel);
```

Visual Basic Declaration

```
    Private Declare Function RACInitCtrl Lib "racw32.dll" _
                            (ByVal hwnd As Long, _
                             ByVal userid As String, _
                             ByVal sysid As String, _
                             ByVal loadMenu As Long, _
                             ByVal loadAccel As Long) As Long
```

typedef

```
typedef int (_RACAPIPTR RACINITCTRL)
                            (HWND hwnd,
                            char far *userid,
                            char far *sysid,
                            BOOL accel,
                            BOOL menu);
```

Calls

RACGetStatus, RACInit, RACLoadAccelerators, RACLoadMenu, RACProcessError, RACRestorePath, RACSetStatus, RACWorkingPath

RACLibVersion

Get library version information.

Syntax

FSI_VERSION* RACLibVersion(void)

Remarks

This returns a pointer to a structure containing version information. The structure is composed of four elements: a numeric version value, a character string, a date, and a time.

Returns

This returns a pointer to a structure containing version information.

Files

Include: racapi.h
Source: RACVERSN.C

Code Declarations

```
    FSI_VERSION * _VMMAPI RACLibVersion(void);
Visual Basic Declaration
    Private Declare Function RACLibVersion Lib "racw32.dll" As Long
```

RACLoadIni

Load the INI file.

Syntax

int RACLoadIni(void)

Remarks

Load the INI file. The file name is defaulted to FSIUSER.INI if RACSetIniFile did not supply it.

Files

Include: racapi.h
Source: RACTHRD.C

Calls

RACGetIniFile, RACSetIniFile, RACSetStatus

RACLoadMenu

This function will query the INI file for the MEN.RES compatible file to load.

Syntax

int RACLoadMenu(void)

Remarks

This function will query the INI file for the MEN.RES compatible file to load. The following INI options will be queried in the order shown. Once a valid option has been determined the menu is created but not activated. A call to RACEnableMenu will be required to activate the menu.

Order of search to determine menu to load:

[AltMenu]

File

If not located then use:

[MENU]

File

If not located the file MEN.RES will be attempted.

If no menu is located, none will be created and an error returned.

RACRestoreMenu will restore the original menu. RACTerminate automatically calls this function.

Returns

This returns RAC_SUCCESS upon success or one of the values defined in RACAPI.H.

Files

Include: racapi.h
Source: RACMENU.C

Code Declarations

```
int _RACAPI RACLoadMenu(void);
```

Visual Basic Declaration

```
Private Declare Function RACLoadMenu Lib "racw32.dll" As Long
```

typedef

```
typedef int (_RACAPIPTR RACLOADMENU) (void);
```

Calls

RACGetCmdLineArg, RACSetStatus

RACMain

Remote Start of Entry (MODAL)

Syntax

```
int RACMain(HWND hwnd,
        char* transaction,
        char* company,
        char* lob,
        char* policy,
        char* description,
        char* userid,
        char* sysid,
        int entrymode)
```

Parameter	Description
hwnd	The external application's main (highest level) window.
transaction	A null-terminated string pointer that represents one of the valid [TRANSACTIONS] abbreviations defined in the INI file. For instance: "NB" for New Business; "EN" for Endorsement; and so on.
company	A null-terminated string pointer that corresponds to the first WIP key value (which is also the first component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Company" value.
lob	A null-terminated string pointer that corresponds to the second WIP key value (which is also the second component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Line of Business" value.
policy	A null-terminated string pointer that corresponds to the WIP KeyID value. This parameter is sometimes referred to as the "ID Number" value.
description	A null-terminated string pointer that should be assigned as the WIP description value.
userid	A null-terminated string pointer that corresponds to a valid user ID with access to the entry system.
sysid	Future expansion - not currently used.
entrymode	A valid RACLib action. Currently defined: AFEACTION_CREATE = Create new WIP. AFEACTION_UPDATE = Update existing WIP.

Remarks

Starts a modal session that is similar to starting AFEMAIN.EXE. No other initialization or termination functions need be called by the user's application. The function does not return until the session completes.

This modal session will take over the main window handling and replace the existing menu. The new menu will be created from the file specified by the [ALTMENU] setting in the INI file. If omitted, the menu will default to the file specified by the [MENU] setting in the INI file. The original menu will be restored before returning.

During initialization, the login screen can appear if the user ID is not specified by the parameter and is not specified in the INI file.

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACMAIN.C

Code Declarations

```
int  _RACAPI RACMain(HWND      hwnd,
                     char far *transaction,
                     char far *company,
                     char far *lob,
                     char far *policy,
                     char far *description,
                     char far *userid,
                     char far *sysid,
                     int       mode);
```

Visual Basic Declaration

```
Private Declare Function RACMain Lib "racw32.dll" _
            (ByVal hwnd As Long, _
             ByVal Key1 As String, _
             ByVal Key2 As String, _
             ByVal KeyID As String, _
             ByVal description As String, _
             ByVal userid As String, _
             ByVal sysid As String, _
             ByVal modeval As Long) As Long
```

typedef

```
typedef void (_RACAPIPTR RACMAIN)(HWND hwnd,
                                  char *transaction,
                                  char *company,
                                  char *lob,
                                  char *policy,
                                  char *description,
                                  char *userid,
                                  char *sysid,
                                  long mode);
```

Example

```
The following code is an excerpt from an external program.
  switch ( msg ) {
  case WM_COMMAND:
      switch(Param1) {
      case ID_CREATE_NEWPOL:
          rval = RACMain( hwnd,
                          "NB",
                          "ACME INSURANCE",
                          "HEALTH",
                          "1402001",
                          "New Sample Policy",
                          "USER1",
                          "",
```

```
                    AFEACTION_CREATE);
        if (rval != RAC_SUCCESS){
                ... // handle error
                break;
        }
        ... // continue success
        break;
    }
```

See Also

RACInitAll, RACCreate, RACUpdate, RACTerminate, RACModal

Calls

RACCreate, RACEnableMenu, RACGetStatus, RACInitAll, RACModal, RACProcessError, RACRestoreMenu, RACUpdate

RACMainWndProc

The Documaker Workstation message handler.

Syntax

MRESULT RACMainWndProc(HWND hwnd,
UINT msg,
WPARAM mp1,
LPARAM mp2)

Parameter	Description
hwnd	Handle of window.
msg	Message
mp1	Message parameter one.
mp2	Message parameter two.

Remarks

Is a main window message handler used by the entry system. During a non-modal session, it is possible to pass messages to this function that should be handled by the entry system. Most often, this function will be passed WM_COMMAND messages associated with menu commands. You must initialize the system before you call this function.

The prototype for this function should conform to that required for a standard window within the operating environment.

Returns

The return value will be that returned from a WndProc in the given operating environment.

Files

Include: racapi.h
Source: RACPROC.C

Code Declarations

```
MRESULT EXPENTRY RACMainWndProc(HWND hwnd,
                                MMSG msg,
                                MPARAM1 mp1,
                                MPARAM2 mp2);
```

Visual Basic Declaration

```
Private Declare Function RACMainWndProc Lib "racw32.dll" _
            (ByVal hwnd As Long, _
             ByVal msg As Long, _
             ByVal mp1 As Long, _
             ByVal mp2 As Long) As Long
```

Example

```
The following code is an excerpt from an external program.
```

```
switch ( msg ) {
case WM_COMMAND:
    rval =  RACMainWndProc( hwnd, msg, Param1, Param2);
    ...
```

See Also

RACInit

Calls

RACGetCmdLineArg, RACGetSession, RACSaveAsProxy, RACSetDesktop, RACSetFocus, RACTermSession

RACModal

Start Modal Entry.

Syntax

int RACModal(void)

Remarks

This causes the session to take over the main window handling. You must initialize the system before you call this function.

This function replaces the existing menu with a new menu if one was loaded. The original menu will be restored before returning.

The function does not return until the Entry system completes. The entry system is terminated but the session is not terminated.

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACMODAL.C

Code Declarations

```
int  _RACAPI RACModal(void);
```

Visual Basic Declaration

```
Private Declare Function RACModal Lib "racw32.dll" As Long
```

Example

The following code is an excerpt from an external program.

```
if ( RACInitAll(hwnd, NULL, NULL) != RAC_SUCCESS ) {
    RACProcessError(RACGetStatus());
    return(rval);
}
RACModal();
```

See Also

RACInit

Calls

RACModalEx

RACPackDatabase

Pack the database.

Syntax

int RACPackDatabase(HWND hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

Packs the database associated with the window session.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
        int _RACAPI RACPackDatabase(HWND hwnd);
    Visual Basic Declaration
        Private Declare Function RACPackDatabase Lib "racw32.dll" _
                    (ByVal hwnd As Long) As Long
```

Calls

RACGetSession

RACRemoveScrollbar

Remove the scroll bar from window.

Syntax

void RACRemoveScrollbar(HWND hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

Remove the scroll bar from window.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
void RACRemoveScrollbar(HWND hwnd);
```

Visual Basic Declaration

```
Private Declare Sub RACRemoveScrollbar Lib "racw32.dll" _
                (ByVal hwnd As Long)
```

typedef

```
typedef void (_RACAPIPTR RACREMOVESCROLLBAR) (HWND hwnd);
```

RACRestoreMenu

Restore the Original Menu.

Syntax

int RACRestoreMenu(void)

Remarks

This will cause the session to restore the original main window's menu if it was replaced by a call to RACEnableMenu. You must initialize the system before you call this function.

RACTerminate automatically calls this function.

Returns

This returns RAC_SUCCESS upon success or one of the values defined in RACAPI.H.

Files

Include: racapi.h
Source: RACMENU.C

Code Declarations

```
int _RACAPI RACRestoreMenu(void);
```

Visual Basic Declaration

```
Private Declare Function RACRestoreMenu Lib "racw32.dll" As Long
```

typedef

```
typedef int (_RACAPIPTR RACRESTOREMENU)(void);
```

RACRestorePath

Restore original working directory.

Syntax

void RACRestorePath(void)

Remarks

This function will restore the working directory to the original one saved by calling RACWorkingPath.

Files

Include: racapi.h
Source: RACINIT.C

Code Declarations

```
    void _RACAPI RACRestorePath(void);
Visual Basic Declaration
    Private Declare Sub RACRestorePath Lib "racw32.dll"
```

See Also

RACWorkingPath

RACRetrieve

Display transaction from archive.

Syntax

```
int RACRetrieve(HWND hwnd,
        char* Key1,
        char* Key2,
        char* KeyID,
        int startpage)
```

Parameter	Description
hwnd	The parent window to contain this application.
Key1	Search data for Key1.
Key2	Search data for Key2.
KeyID	Search data for KeyID.
startpage	Page to begin display of form set.

Remarks

Display transaction from archive, similar to RACRetrieveArchive except that it does not call RACInit, RACInit must be called before this function.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
int _RACAPI RACRetrieve(HWND hwnd,
                        char *Key1,
                        char *Key2,
                        char *KeyID,
                        int startpage);
```

Visual Basic Declaration

```
Private Declare Function RACRetrieve Lib "racw32.dll" _
                (ByVal hwnd As Long, _
                 ByVal Key1 As String, _
                 ByVal Key2 As String, _
                 ByVal KeyID As String, _
                ByVal startpage As Long) As Long
```

Calls

RACFilterPageForDisplay, RACProcessError, RACSetCurSession, RACSetStatus

RACRetrieveArchive

View form set specified.

Syntax

HWND RACRetrieveArchive(HWND hwndParent,
char* Key1,
char* Key2,
char* KeyID,
int startpage,
int location)

Parameter	Description
hwndParent	The parent window to contain this application.
Key1	Search data for Key1.
Key2	Search data for Key2.
KeyID	Search data for KeyID.
startpage	set beginning page
location	Set screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

It is assumed that the window's handle passed represents another application's main window. We'll create a frame window to contain the form set view and start the entry system after usual initialization.

See the <ArcRet> INI group to determine which fields are matched with Key1, Key2, and KeyID.

Returns

If successful, the handle of our desktop window is returned. On failure, zero will be returned and the caller should use RACGetStatus to retrieve the last know error code. Error code values are defined in RACAPI.H.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
HWND  _RACAPI RACRetrieveArchive(HWND hwndParent,
                                 char *Key1,
                                 char *Key2,
                                 char *KeyID,
```

```
                              int startpage,
                              int location);
```

Visual Basic Declaration

```
    Private Declare Function RACRetrieveArchive Lib "racw32.dll" _
                    (ByVal hwndParent As Long, _
                     ByVal Key1 As String, _
                     ByVal Key2 As String, _
                     ByVal KeyID As String, _
                     ByVal startpage As Long, _
                     ByVal location As Long) As Long
```

Calls

RACRetrieveArchiveStat

RACRetrieveArchiveHab

View form set specified.

Syntax

HWND RACRetrieveArchiveHab(HINSTANCE hab,
char* Key1,
char* Key2,
char* KeyID,
int startpage,
int location)

Parameter	Description
hab	Anchor block. In Windows, it is the instance handle.
Key1	Search data for Key1.
Key2	Search data for Key2.
KeyID	Search data for KeyID.
startpage	set beginning page
location	Set screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

Is identical to RACViewByKey except the application instance is passed instead of a window's handle. Actually, RACRetrieveArchiveHab uses the window's handle only to obtain the application instance.

Returns

If successful, the handle of our desktop window is returned. On failure, zero will be returned and the caller should use RACGetStatus to retrieve the last know error code. Error code values are defined in RACAPI.H.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
HWND  _RACAPI RACRetrieveArchiveHab(HAB hab,
                                    char *Key1,
                                    char *Key2,
                                    char *KeyID,
                                    int startpage,
                                    int location);
```

Visual Basic Declaration

```
Private Declare Function RACRetrieveArchiveHab Lib "racw32.dll" _
                (ByVal hInstance As Long, _
                 ByVal Key1 As String, _
                 ByVal Key2 As String, _
                 ByVal KeyID As String, _
                 ByVal startpage As Long, _
                 ByVal location As Long) As Long
```

Calls

RACRetrieveArchiveStat

RACSave

Save transaction.

Syntax

int RACSave(void)

Remarks

Save the current transaction returns SUCCESS if the transaction was saved or if no transaction was current and needed to be saved.

Files

Include: racapi.h
Source: RACTERM.C

Code Declarations

```
    int _RACAPI RACSave(void);
```

Visual Basic Declaration

```
    Private Declare Function RACSave Lib "racw32.dll" () As Long
```

typedef

```
typedef int (_RACAPIPTR RACSAVE) (void);
```

Calls

RACGetSession

RACSaveAsProxy

Save the current form set in the current AFEData structure into a proxy file.

Syntax

int RACSaveAsProxy(HINSTANCE hab,
HWND hwnd,
VMMHANDLE menuH)

Parameter	Description
hab	Anchor block or instance handle for Windows.
hwnd	Handle of window.
menuH	Menu handle.

Remarks

Save the current form set in the current AFEData structure into a proxy file.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
    int _VMMAPI RACSaveAsProxy(HAB hab, HWND hwnd, VMMHANDLE menuH);
Visual Basic Declaration
    Private Declare Function RACSaveAsProxy Lib "racw32.dll" _
                    (ByVal hInstance As Long, _
                     ByVal hWnd As Long, _
                     ByVal menuH As Long) As Long
```

Calls

RACGetSession

RACSetCaptionOff

Set caption off for this window.

Syntax

void RACSetCaptionOff(HWND hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

Set caption off for this window.

Files

Include: racapi.h
Source: RACPROC.C

Code Declarations

```
        void _RACAPI RACSetCaptionOff(HWND hwnd);
    Visual Basic Declaration
        Private Declare Sub RACSetCaptionOff Lib "racw32.dll" _
                        (ByVal hwnd As Long)
```

typedef

```
    typedef void (_RACAPIPTR RACSETCAPTIONOFF) (HWND hwnd);
```

Calls

RACGetSession

RACSetCaptionOn

Set caption on for this window.

Syntax

void RACSetCaptionOn(HWND hwnd)

Parameter	Description
hwnd	Handle of window.

Remarks

Set caption on for this window.

Files

Include: racapi.h
Source: RACPROC.C

Code Declarations

```
    void _RACAPI RACSetCaptionOn(HWND hwnd);
```

Visual Basic Declaration

```
    Private Declare Sub RACSetCaptionOn Lib "racw32.dll" _
                    (ByVal hwnd As Long)
```

typedef

```
typedef void (_RACAPIPTR RACSETCAPTIONON) (HWND hwnd);
```

Calls

RACGetSession

RACSetCmdLineArg

This allows historical command line options for AFEMAIN program to be passed to RACInit.

Syntax

```
int RACSetCmdLineArg(int argc,
                     char* argv[ ])
```

Parameter	Description
argc	Number of parameters.
argv	Array of pointers that contain the parameters.

Remarks

This allows historical command line options for AFEMAIN program to be passed to RACInit. Parameters are identical to those of a c program's main function.

The pointer to the structure is initially NULL but it's filled by the RACSetCmdLineArg function. The memory for the structure is static but there is a linked list's handle that is created via VMMCreateList. This handle will not survive RACTerminate. Therefore, RACSetCmdLineArg will need to be called before each RACInit.

Returns

SUCCESS or FAIL

Files

Include: racapi.h
Source: RACCMDL.C

Code Declarations

```
    int _RACAPI RACSetCmdLineArg(int argc, char *argv[]);
```

Visual Basic Declaration

```
    Private Declare Function RACSetCmdLineArg Lib "racw32.dll" _
                    (ByVal argc As Long, _
                     ByVal argv As Long) As Long
```

typedef

```
typedef int (_RACAPIPTR RACSETCMDLINEARG)
                        (int argc, char *argv[]);
```

Example

```
Valid command line options

    -INI or /INI - set the INI FILE.
    -MODE or /MODE - set to one of the following to etrieve WIP
    -USERID or /USERID - set the user ID
    -HD or /HD - set help debug flag
    -MENU or /MENU - set the menu file.
```

Calls

RACSetIniFile

RACSetCurSession

Set this window to be the current session.

Syntax

int RACSetCurSession(HWND hwnd)

Parameter	Description
hwnd	Handle of window

Remarks

Set the main frame window's handle (hwndMainFrame variable) if we have a valid session for the handle passed.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
int _RACAPI RACSetCurSession(HWND hwnd);
```

Visual Basic Declaration

```
Private Declare Function RACSetCurSession Lib "racw32.dll" _
                (ByVal hWnd As Long) As Long
```

typedef

```
typedef int (_RACAPIPTR RACSETCURSESSION)(HWND hwnd);
```

Calls

RACGetSession, RACSetStatus

RACSetHook

Establish Windows message hook.

Syntax

void RACSetHook(void)

Remarks

This installs RACHookProc as a Windows compatible message hook. See that function for more information.

Files

Include: racapi.h
Source: RACHOOK.C

Code Declarations

```
    void _RACAPI RACSetHook(void);
Visual Basic Declaration
    Private Declare Sub RACSetHook Lib "racw32.dll" ()
```

RACSetIniFile

Sets the INI file path name used by RACLib functions.

Syntax

int RACSetIniFile(char* iniFile)

Parameter	Description
iniFile	File path name of INI file.

Remarks

Sets the INI file path name used by RACLib functions. A string is set within RACLib that is used by the RACInit function.

Returns

RAC_SUCCESS (0) is always returned.

Files

Include: racapi.h
Source: RACINIT.C

Code Declarations

```
    int _RACAPI RACSetIniFile(char *iniFile);
Visual Basic Declaration
    Private Declare Function RACSetIniFile Lib "racw32.dll" _
                    (ByVal iniFile As String) As Long
```

typedef

```
typedef int (_RACAPIPTR  RACSETINIFILE) (char *iniFile);
```

See Also

RACGetIniFile

RACSetParent

Use this function to set the parent window. This function lets you size the frame window which contains the form sets you display with these functions:

- RACViewData
- RACViewDataHab
- RACViewByKey
- RACViewByKeyHab
- RACRetrieveArchive
- RACRetrieveArchiveHab

If you omit the RACSetParent function, the size of the frame window is 1/3 of the screen size. In some cases, the text on the form set may be too small to read at this size.

Keep in mind that the form must be open before you can set it as a parent.

Syntax

void RACSetParent(HWND hwnd)

Remarks

Set the parent window for RACThread if the parent window has been set and is a valid window's handle then we create a child window instead of using the desktop.

Files

Include: racapi.h
Source: RACTHRD.C

Code Declarations

```
    void _RACAPI RACSetParent(HWND hwnd);
Visual Basic Declaration
    Private Declare Sub RACSetParent Lib "racw32.dll" _
                (ByVal hwnd As Long)
```

typedef

```
typedef void (_RACAPIPTR RACSETPARENT) (HWND hwnd);
```

RACSetProxyHandle

Set the document handle of a proxy file for RACViewProxy.

Syntax

void RACSetProxyHandle(VMMHANDLE docH)

Remarks

Set the document handle of a proxy file for RACViewProxy.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
    void _RACAPI RACSetProxyHandle(VMMHANDLE docH);
Visual Basic Declaration
    Private Declare Sub RACSetProxyHandle Lib "racw32.dll" _
                    (ByVal docH As Long)
```

typedef

```
typedef void (_RACAPIPTR RACSETPROXYHANDLE)
                         (VMMHANDLE docH);
```

RACSetSessionMenu

Set the menu's handle in the session structure

Syntax

unsigned short RACSetSessionMenu(VMMHANDLE hwnd,
VMMHANDLE newMenuH)

Parameter	Description
hwnd	Menu Handle.
newMenuH	New Menu handle.

Remarks

This should be a call back function from FWMSetCurrentMenu in GUILIB.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
WORD _VMMAPI RACSetSessionMenu(VMMHANDLE oldMenuH,
                               VMMHANDLE newMenuH);
```

Visual Basic Declaration

```
Private Declare Function RACSetSessionMenu Lib "racw32.dll" _
                (ByVal oldMenuH As Long, _
                 ByVal newMenuH As Long) As Long
```

Calls

RACGetFirstSession, RACGetNextSession

RACSetStatus

Set the Error Status Code

Syntax

int RACSetStatus(int status)

Parameter	Description
status	Status code

Remarks

Assigns the current error code that will be returned via RACGetStatus. Normally, this function will be called by functions within RACLib, but can be called by the master application during a non-modal session.

Returns

The value returned is the same as the parameter passed. This usually will be one of the values defined in the RACAPI.H file.

Files

Include: racapi.h
Source: RACDATA.C

Code Declarations

```
int  _RACAPI RACSetStatus(int sts);
```

Visual Basic Declaration

```
Private Declare Function RACSetStatus Lib "racw32.dll" _
                (ByVal stsCode As Long) As Long
```

typedef

```
typedef int (_RACAPIPTR RACSETSTATUS) (int status);
```

RACSetWorkingPath

Set current working directory.

Syntax

void RACSetWorkingPath(char* path)

Parameter	Description
path	path to make the current directory

Remarks

Set current working directory to whatever is in the parameter. The old parameter is saved and can be restored by the RACRestorePath function.

Files

Include: racapi.h
Source: RACINIT.C

Code Declarations

```
    void _RACAPI RACSetWorkingPath(char *path);
```

Visual Basic Declaration

```
    Private Declare Sub RACSetWorkingPath Lib "racw32.dll" _
                        (ByVal filename As String)
```

typedef

```
typedef int (_RACAPIPTR RACSETWORKINGPATH) (char *path);
```

RACSubClass

Subclass a window's procedure with RACMainWndProc.

Syntax

PFNWP RACSubClass(HWND hwnd)

Parameter	Description
hwnd	Handle of window

Remarks

Subclass window specified with RACMainWndProc.

Returns

A pointer to the original window procedure is returned. If error, then NULL is returned.

Files

Include: racapi.h
Source: RACTHRD.C

Code Declarations

```
PFNWP _RACAPI RACSubClass(HWND hwnd);
```

Visual Basic Declaration

```
Private Declare Function RACSubClass Lib "racw32.dll" _
                (ByVal hwnd As Long) As Long
```

RACTerminate

Terminate Entry Session.

Syntax

int RACTerminate(unsigned int mode)

Parameter	Description
mode	The parameter mode should be TRUE if the entry system should ask the user to save (if necessary) and FALSE to cause the system to shutdown without asking.

Remarks

This will cause the session to release all resources used by the entry system. You must initialize the system before you call this function.

> A non-modal session relies upon the controlling application to request termination. Exiting without terminating the system can cause unpredictable results.

Returns

Upon success, returns RAC_SUCCESS or one of the values defined in RACAPI.H.

Files

Include: racapi.h
Source: RACTERM.C

Code Declarations

```
int  _RACAPI RACTerminate(BOOL mode);
```

Visual Basic Declaration

```
Private Declare Function RACTerminate Lib "racw32.dll" _
                (ByVal modeval As Long) As Long
```

typedef

```
typedef int   (_RACAPIPTR RACTERMINATE) (BOOL flag);
             typedef int (*RACTERMFUNC)(BOOL mode);
```

Calls

RACClearSessions, RACGetFirstSession, RACGetNextSession, RACInitData, RACResetCmdLineArg, RACRestorePath, RACTermSession, RACUnhook

RACTermSession

Terminate Entry Session.

Syntax

int RACTermSession(unsigned int mode,
HWND hwnd)

Parameter	Description
mode	The parameter mode should be TRUE if the entry system should ask the user to save (if necessary) and FALSE to cause the system to shutdown without asking.
hwnd	Handle of window

Remarks

This will cause the session to release all resources used by the entry system. You must initialize the system before you call this function.

> A non-modal session relies upon the controlling application to request termination. Exiting without terminating the system can cause unpredictable results.

Returns

Upon success, returns RAC_SUCCESS or one of the values defined in RACAPI.H.

Files

Include: racapi.h
Source: RACTERM.C

Code Declarations

```
int _RACAPI RACTermSession(BOOL mode, HWND hwnd);
```

Visual Basic Declaration

```
Private Declare Function RACTermSession Lib "racw32.dll" _
                (ByVal modeval As Long, _
                 ByVal hwnd As Long) As Long
```

typedef

```
typedef void (_RACAPIPTR RACTERMSESSION)
                          (BOOL mode, HWND hwnd);
```

Calls

RACGetSession, RACReleaseSession, RACSetParent

RACThread

New Process operating within another window.

Syntax

int RACThread(HWND hwndParent)

Parameter	Description
hwndParent	The parent window to contain this application.

Remarks

In this entry point, it is assumed that the window's handle passed represents the area that must contain the program. We'll create a frame window that we are familiar with inside this area and start the entry system after usual initialization.

Returns

If successful, RAC_SUCCESS is returned otherwise a value from RACAPI.H is returned.

Files

Include: racapi.h
Source: RACTHRD.C

Code Declarations

```
    int  _RACAPI RACThread( HWND hwndParent );
Visual Basic Declaration
    Private Declare Function RACThread Lib "racw32.dll" _
                    (ByVal hwndParent As Long) As Long
```

typedef

```
typedef int   (_RACAPIPTR RACTHREAD) (HWND hwnd);
```

See Also

RACThread, RACInitAll, RACSetHook, RACEnableMenu

Calls

RACCreateMainWindow, RACEnableMenu, RACGetDescription, RACGetIniFile, RACGetStatus, RACInitAll, RACLoadMenuToolbar, RACPostInit, RACProcessError, RACSetHook, RACSetIniFile, RACSetStatus

RACUnhook

Remove Windows message hook.

Syntax

void RACUnhook(void)

Remarks

This De-installs RACHookProc as a Windows compatible message hook. See that function for more information.

Files

Include: racapi.h
Source: RACHOOK.C

Code Declarations

```
    void _RACAPI RACUnhook(void);
Visual Basic Declaration
    Private Declare Sub RACUnhook Lib "racw32.dll" ()
```

RACUpdate

Edit an existing WIP entry.

Syntax

```
extern int RACUpdate(char* transaction,
                     char* company,
                     char* lob,
                     char* policy,
                     char* description,
                     int entryMode)
```

Parameter	Description
transaction	A null-terminated string pointer that represents one of the valid [TRANSACTIONS] abbreviations defined in the INI file. For instance: "NB" for New Business; "EN" for Endorsement; and so on.
company	A null-terminated string pointer that corresponds to the first WIP key value (which is also the first component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Company" value.
lob	A null-terminated string pointer that corresponds to the second WIP key value (which is also the second component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Line of Business" value.
policy	A null-terminated string pointer that corresponds to the WIP KeyID value. This parameter is sometimes referred to as the "ID Number" value.
description	A null-terminated string pointer that should be assigned as the WIP description value.
entryMode	AFEACTION_UPDATE - Update existing WIP.

Remarks

Attempts to reload for edit an existing WIP entry during a non-modal session. You must initialize the system before you call this function.

Returns

RAC_SUCCESS (0) is returned if the session completes successfully, otherwise one of the values defined in RACAPI.H is returned.

Files

Include: racapi.h
Source: RACUPDAT.C

Code Declarations

```
int  _RACAPI RACUpdate( char far *transaction,
                        char far *company,
                        char far *lob,
                        char far *policy,
```

```
                            char far *description,
                            int       mode);
```

Visual Basic Declaration

```
    Private Declare Function RACUpdate Lib "racw32.dll" _
                    (ByVal transaction As String, _
                     ByVal Key1 As String, _
                     ByVal Key2 As String, _
                     ByVal KeyID As String, _
                     ByVal description As String, _
                     ByVal mode As Long) As Long
```

Example

The following code is an excerpt from an external program.

```
    switch ( msg ) {
    case WM_COMMAND:
        switch(Param1){
        case ID_UPDATE_OLDPOL:
            rval = RACUpdate("NB",
                             "ACME INSURANCE",
                             "HEALTH",
                             "1402001",
                             "New Sample Policy",
                             AFEACTION_UPDATE);
            if (rval != RAC_SUCCESS){
                ...  // handle error
                break;
            }
            ... // continue success
            break;
        }
```

See Also

RACInit

Calls

RACWipSelection

RACViewByKey

View form set specified.

Syntax

HWND RACViewByKey(HWND hwndParent,
char* key,
int startpage,
int location)

Parameter	Description
hwndParent	The parent window to contain this application.
key	The archive's key for the transaction.
startpage	The page number to start display (defaults to 1)
location	Screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

It is assumed that the window's handle passed represents another application's main window. We'll create a frame window to contain the form set view and start the entry system after usual initialization.

Returns

If successful, the handle of our desktop window is returned. On failure, zero will be returned and the caller should use RACGetStatus to retrieve the last know error code. Error code values are defined in RACAPI.H.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
HWND _RACAPI RACViewByKey( HWND hwndParent,
                           char *key,
                           int startpage,
                           int location );
```

Visual Basic Declaration

```
Private Declare Function RACViewByKey Lib "racw32.dll" _
                (ByVal hwndParent As Long, _
                 ByVal key As String, _
                 ByVal startpage As Long, _
                 ByVal location As Long) As Long
```

typedef

```
typedef HWND (_RACAPIPTR RACVIEWBYKEY)
                            (HAB hab,
                            char *key,
                            int startpage,
                            int location);
```

See Also

RACCreateMainWindow, RACInitAll

Calls

RACViewByKeyStat

RACViewByKeyHab

View form set specified.

Syntax

HWND RACViewByKeyHab(HINSTANCE hhab,
char* key,
int startpage,
int location)

Parameter	Description
hhab	Anchor block or instance handle for Windows.
key	The archive's key for the transaction.
startpage	The page number to start display (defaults to 1)
location	Screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

Is identical to RACViewByKey except the application instance is passed instead of a window's handle. Actually, RACViewData uses the window's handle only to obtain the application instance.

Returns

If successful, the handle of our desktop window is returned. On failure, zero will be returned and the caller should use RACGetStatus to retrieve the last know error code. Error code values are defined in RACAPI.H.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
HWND  _RACAPI RACViewByKeyHab( HAB hhab,
                               char *key,
                               int startpage,
                               int location );
```

Visual Basic Declaration

```
Private Declare Function RACViewByKeyHab Lib "racw32.dll" _
                (ByVal hInstance As Long, _
                 ByVal key As String, _
                 ByVal startpage As Long, _
                 ByVal location As Long) As Long
```

typedef

```
typedef HWND (_RACAPIPTR RACRETRIEVEARCHIVE)
                            (HAB hab,
                            char *Key1,
                            char *Key2,
                            char *KeyID,
                            int startpage,
                            int location);
```

See Also

RACCreateMainWindow, RACInitAll

Calls

RACViewByKeyStat

RACViewData

View form set specified.

Syntax

HWND RACViewData(HWND hwndParent,
char* filename,
int startpage,
int location)

Parameter	Description
hwndParent	The parent window to contain this application.
filename	The name of a file to import.
startpage	The page number to start display (defaults to 1)
location	Screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

It is assumed that the window's handle passed represents another application's main window. We'll create a frame window to contain the form set view and start the entry system after usual initialization.

Returns

If successful, the handle of our desktop window is returned. On failure, zero will be returned and the caller should use RACGetStatus to retrieve the last know error code. Error code values are defined in RACAPI.H.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
HWND  _RACAPI RACViewData( HWND hwndParent,
                           char *filename,
                           int startpage,
                           int location );
```

Visual Basic Declaration

```
Private Declare Function RACViewData Lib "racw32.dll" _
                (ByVal hwndParent As Long, _
                 ByVal filename As String, _
                 ByVal startpage As Long, _
                 ByVal location As Long) As Long
```

typedef

```
typedef int (*RACVIEWFUNC)(HWND hwnd,
                           char *filename,
                           int startpage);
```

See Also

RACCreateMainWindow, RACInitAll

Calls

RACViewDataStat

RACViewDataHab

View form set specified.

Syntax

HWND RACViewDataHab(HINSTANCE hhab,
char* filename,
int startpage,
int location)

Parameter	Description
hhab	Anchor block or instance handle for Windows.
filename	The name of a file to import.
startpage	The page number to start display (defaults to 1)
location	Screen location. 0 = Left half display (default) 1 = Right half display 2 = Top half display 3 = Bottom half display 4 = normal window 5 = maximized window

Remarks

Is identical to RACViewData except the application instance is passed instead of a window's handle. Actually, RACViewData uses the window's handle only to obtain the application instance.

Returns

TRUE or FALSE (0) is returned depending upon whether the record matches the search criteria.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
HWND  _RACAPI RACViewDataHab( HAB hhab,
                              char *filename,
                              int startpage,
                              int location );
```

Visual Basic Declaration

```
Private Declare Function RACViewDataHab Lib "racw32.dll" _
                (ByVal hInstance As Long, _
                 ByVal filename As String, _
                 ByVal startpage As Long, _
                 ByVal location As Long) As Long
```

typedef

```
typedef HWND (_RACAPIPTR RACVIEWDATA)
                            (HAB hab,
                            char *filename,
                            int startpage,
                            int location);
```

Calls

RACViewDataStat

RACViewProxy

Display archive from proxy file.

Syntax

int RACViewProxy(HWND hwnd,
char* filename,
int startpage)

Parameter	Description
hwnd	must be a valid window's handle to contain form set display
filename	path to a valid proxy file
startpage	page to begin display of form set

Remarks

Display archive from proxy file, RACInit must be called before this function.

Files

Include: racapi.h
Source: RACVIEW.C

Code Declarations

```
int _RACAPI RACViewProxy(HWND hwnd,
                         char *filename,
                         int startpage);
```

Visual Basic Declaration

```
Private Declare Function RACViewProxy Lib "racw32.dll" _
                (ByVal hwnd As Long, _
                 ByVal filename As String, _
                 ByVal startpage As Long) As Long
```

typedef

```
typedef int (_RACAPIPTR RACVIEWPROXY)
                        (HWND hwnd,
                        char *filename,
                        int startpage);
```

Calls

RACFilterPageForDisplay, RACGetDescription, RACGetIniFile, RACGetStatus, RACProcessError, RACSetCurSession, RACSetStatus

RACWipSelectFunction

WIP selection support function.

Syntax

unsigned int RACWipSelectFunction(void* WipRec)

Parameter	Description
WipRec	Pointer to a WIP record that is to be evaluated.

Remarks

This is a support function for RACUpdate. The WIP record is first evaluated by the AFEWipSelectFunction to see if it is a valid record for this user. A second test is then performed via RACWipSelection to see if the record matches the remaining search criteria.

Returns

TRUE or FALSE (0) is returned depending upon whether the record matches the search criteria.

Files

Include: racapi.h
Source: RACUPDAT.C

Code Declarations

```
BOOL _VMMAPI RACWipSelectFunction(void FAR *WipRec);
```

Visual Basic Declaration

```
Cannot be called from VB
```

Calls

RACWipSelection

RACWipSelection

Support function for WIP selection

Syntax

unsigned int RACWipSelection(void* WipRec,
char* transaction,
char* company,
char* lob,
char* policy)

Parameter	Description
WipRec	Pointer to a WIP record that is to be evaluated.
transaction	A null-terminated string pointer that represents one of the valid [TRANSACTIONS] abbreviations defined in the INI file. For instance: "NB" for New Business; "EN" for Endorsement; and so on.
company	A null-terminated string pointer that corresponds to the first WIP key value (which is also the first component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Company" value.
lob	A null-terminated string pointer that corresponds to the second WIP key value (which is also the second component of a line from the FORM.DAT). This parameter is sometimes referred to as the "Line of Business" value.
policy	A null-terminated string pointer that corresponds to the WIP KeyID value. This parameter is sometimes referred to as the "ID Number" value.

Remarks

This is a support function for RACUpdate used to compare a WIP record for matches search criteria.

When a NULL WipRec parameter is used, the remaining parameters are saved as the compare test information. Afterwards the function can be called with valid WipRec parameters to test for matches.

When a valid WipRec parameter is used, the remaining parameters are not used. Rather, the WIP record is compared to the static test information stored previously.

Returns

TRUE or FALSE (0) is returned depending upon whether the record matches the search criteria.

Files

Include: racapi.h
Source: RACUPDAT.C

Code Declarations

```
BOOL _RACAPI RACWipSelection(void far *WipRec,
                             char far *transaction,
                             char far *company,
                             char far *lob,
```

```
                                char far *policy);
```

Visual Basic Declaration

```
    Private Declare Function RACUpdate Lib "racw32.dll" _
                    (ByVal WipRec As Long, _
                     ByVal transaction As String, _
                     ByVal Key1 As String, _
                     ByVal Key2 As String, _
                     ByVal KeyID As String) As Long
```

RACWorkingPath

Establish correct working directory.

Syntax

void RACWorkingPath(void)

Remarks

This function queries the environment for FSIPATH. If a path is specified, the original path is saved and the working directory is changed to the one specified by the environment variable. Calling RACRestorePath can restore original working directory.

If no FSIPATH environment variable is available, the working directory is not changed.

Files

Include: racapi.h
Source: RACINIT.C

Code Declarations

```
    void _RACAPI RACWorkingPath(void);
Visual Basic Declaration
    Private Declare Sub RACWorkingPath Lib "racw32.dll" ()
```

See Also

RACRestorePath

Calls

RACSetWorkingPath

Global Types

CMDLINE

typedef struct CMDLINE

This structure contains information parsed from the command line that is used by RACInit.

Files

Source: RACAPI.H

struct CMDLINE

Name	Description
char ini[129 + 1]	The INI file name supplied by the /INI command line switch. INI=\fap\mstrres\sampco\fsiuser.ini
int gMode	The mode state supplied by the /MODE command line switch. MODE=retrieve MODE=WIP
char CmdLineUserID[129 + 1]	The user ID set by the /USERID command line switch. USERID="FORMAKER"
BOOL bHelpDebug	True if the help system debug /HD command line switch is supplied.
char menufile[129 + 1]	The menu file name supplied by the MENU command line switch. MENU=men.res
VMMHANDLE argvH	Handle to argument list.
char progName[129 + 1]	Program name.

Files

Source: RACAPI.H

MRESULT

typedef long MRESULT

Files

Source: RACAPI.H

PFNWP

typedef WNDPROC PFNWP

Files

Source: RACAPI.H

RAC_ENTRYMODES

typedef enum RAC_ENTRYMODES

Files

Source: RACAPI.H

enum RAC_ENTRYMODES

Name	Description
AFEACTION_CREATE=1	Successful operation
AFEACTION_UPDATE=2	The main window handle is invalid or 0

Files

Source: RACAPI.H

RAC_ERRNO

typedef enum RAC_ERRNO

Error codes supplied as return values and are used in the RACSetStatus and RACGetStatus functions.

Files

Source: RACAPI.H

enum RAC_ERRNO

Name	Description
RAC_SUCCESS=0	Successful operation.
RAC_ERR_HWND=-1	The main window handle is invalid or 0.
RAC_ERR_ACCESS=-2	DLL is currently accessed.
RAC_ERR_FAP=-3	FAP Failed to initialize.
RAC_ERR_AFE=-4	Failed to initialize *AFELib*.
RAC_ERR_MENU=-5	Failed to load menu.
RAC_ERR_MODE=-6	The entry mode specified is not valid.
RAC_ERR_TRANS=-7	Transaction error transaction invalid.
RAC_ERR_COMPANYLOB=-8	Company or Line of Business error.
RAC_ERR_ENTRY=-9	Entry failed to start.
RAC_ERR_WIPCREATE=-10	Cannot create WIP record.
RAC_ERR_WIPSTATUS=-11	WIP status invalid.
RAC_ERR_NEWWIP=-12	WIP *NEW* type not specified in INI.
RAC_ERR_POLICY=-13	Policy number not specified.
RAC_ERR_WIPSEL=-14	WIP selection not completed.
RAC_ERR_FORMSET=-15	Form set load failed.
RAC_ERR_AFEDLL=-16	Cannot locate *AFE* DLL.
RAC_ERR_NOWIP=-17	No WIP to select from.
RAC_ERR_AFEDATA=-18	AFEDATA incorrect.
RAC_ERR_AFEFUNCION=-19	AFE function failed.
RAC_ERR_AFEINI=-20	AFE failed to load INI.
RAC_ERR_NEEDTERM=-21	Session already active.
RAC_ERR_NODLL=-22	Could not load DLL used by CRacLib.
RAC_ERR_MAXSESSIONS=-23	Maximum <R>RACLib sessions.
RAC_ERR_SESSION=-24	Invalid RACLib session requested.
RAC_ERR_MISC=-25	Unknown error in RACLib.

RAC_ERR_CANCEL=-26	User canceled.
RAC_ERR_TERM=-27	Termination error.
RAC_ERR_NOTUNIQUE=-101	WIP Key components are not unique.
RAC_ERR_ARC=-102	Archive file failure.
RAC_ERR_ARCDFD=-103	Archive DFD file failure.
RAC_ERR_WIPADD=-104	WIP failed to add.
RAC_ERR_PROXY=-105	Proxy error.

Files

Source: RACAPI.H

RACDATA

struct RACDATA

Name	Description
HWND hwndMainFrame	handle to the main frame window
int mainWndCreated	Was the main window created?
int loginfailed	Did login fail?
int racUserClosed	Has the user closed?
HHOOK hHook	Hook procedure
char SysID[20]	System Identification.
char CmdLineUserID[129 + 1]	Command line user identification
PAFEDATA pAFEData	AFE data structure.
short RACRetVal	Latest RACLib return value.
HWND hwndMain	The main client window.
HWND hwndHelpInstance	The help instance
int closeBar	Flag indicates whether close bar window was created
int needTerminate	Flag indicates whether we need to call RACTerminate
int terminateAllowed	Flag indicates whether we can terminate the active session
PFNWP oldProc	Old window procedure
PFNWP clientProc	New client window procedure
HWND oldMenu	Old menu handle.
HWND newMenu	New menu handle.
char* m_ErrorMsg	Error message string.
char oldworkdir[(129)]	Old working directory.
long ulDriveNum	Drive number.
int RACTermSession	true after RACClose has executed on this instance.
VMMHANDLE curmenuH	Current menu handle.
int modal	Is this a modal session?
char proxyfilename[(129)]	Proxy file name.
char Title[(129)]	Title for window.
int NoCaptionWindow	Is there a caption?
long orgWindowStyle	Original window style.

Files

Source: RACLIB.H

RACWNDCMDS

typedef enum RACWNDCMDS

Window Commands

Files

Source: RACAPI.H

enum RACWNDCMDS

Name	Description
CMD_ID_ZOOMDLG=1013	Bring up zoom dialog.
CMD_ID_ZOOMIN=1010	Zoom in.
CMD_ID_ZOOMOUT=1011	Zoom out.
CMD_ID_ZOOMNORM=1012	Zoom normal.
CMD_ID_FITWIDTH=1014	Fit to width.
CMD_ID_FITWINDOW=1072	Fit to window.
CMD_ID_PAGETOP=1022	Go to top of page.
CMD_ID_PAGEBOTTOM=1023	Go to bottom of page.
CMD_ID_PAGENEXT=1033	Go to next page.
CMD_ID_PAGEPREV=1034	Go to previous page.
CMD_ID_PAGELEFT=1024	Move page left.
CMD_ID_PAGERIGHT=1025	Move page right.
CMD_ID_FORMFIRST=1016	Go to first form.
CMD_ID_FORMLAST=1017	Go to last form.
CMD_ID_FORMNEXT=1007	Go to next form.
CMD_ID_FORMPREV=1008	Go to previous form.
CMD_ID_LINEUP=1020	Go up a line.
CMD_ID_LINEDOWN=1021	Go down a line.
CMD_ID_REFRESH=1009	Refresh display.
CMD_ID_CASCADE=1002	Cascade windows.
CMD_ID_TILE=1003	Tile windows.
CMD_ID_STACK=1004	Stack windows.
CMD_ID_PRT_FORMSET=1065	Print form set.
CMD_ID_PRT_FORM=1066	Print form.
CMD_ID_PRT_PAGE=1067	Print page.
CMD_ID_PRT_NODLG=1100	Print.

CMD_ID_COUNTPAGES=1	Count pages.

Files

Source: RACAPI.H

VMMHANDLE

typedef void *VMMHANDLE

Files

Source: RACAPI.H