

Developer Manual

RemoteAccess 'C' Developer Toolkit

**'C' Development code & examples for
programming RemoteAccess based BBS's**

© Copyright 1996, *Envy Technologies.*

Contents

Part 1 Introduction

- Introduction to RACDK
- Licence agreement
- Revision history
- Requirements
- Setting up RACDK

Part 2 Using RACDK

- Structures
- Functions
- Pascal string handling
- General tips
- Examples

Appendix

- Frequently asked questions
- File summary
- Contacting Envy Technologies
- Future enhancements
- Other sources
- Credits & acknowledgements

Introduction

Introduction to RACDK

The RemoteAccess 'C' Developer Kit (or RACDK), is a set of files to assist with developing applications for use with the popular bulletin-board package, *RemoteAccess*.

Currently included are the RemoteAccess structures converted to 'C' from Pascal, routines for converting Pascal strings to and from C strings, multitasking timeslicing, CRC calculations and a few examples to get you started.

Licence agreement

The RACDK package consists of all files contained within the distribution archive except for any parts described in **Exclusions** at the end of this section. Before running RACDK, you must read and agree to the following conditions. If you do not or can not agree to and accept the following conditions, you are prohibited from using this package.

- RACDK is copyrighted Damien Guard/Envy Technologies and may only be used in accordance with the conditions set forth in this license agreement.
- RACDK is distributed as a FREEWARE package for this release. This means that the software may be freely distributed and used but Envy Technologies retains all copyrights on this package. This licensing agreement does not apply to any future releases which may be packaged with an alternative licensing agreement.
- Permission is granted to any individual or institution to use, copy, or redistribute RACDK free of charge as long as the distribution package is not modified, nor sold for profit.
- RACDK may be used free of charge in private, shareware and commercial packages but only as part of an executable binary in a compiled state. No files contained in this package can be included as part of your application in their native uncompiled state (.H etc).
- RACDK may not be licenced nor sub-licenced to any third party without written permission of Envy Technologies.
- This licence agreement may not be modified nor supplemented without written permission of Envy Technologies.
- Neither Damien Guard, nor Envy Technologies are obligated to provide new versions or support for existing versions of RACDK even if a previous version of this package is unusable.
- Although care has been taken to write and test a package that does what this document states, the package is provided as is, without warranty or guarantee of any kind, either expressed or implied, as to the quality or performance of this package, except that RACDK will occupy disk space.
- Neither the author of RACDK, Damien Guard, nor the publisher, Envy Technologies are responsible for any (direct or indirect) damage or costs, including, but not limited to, lost savings, lost profits and loss of data, which may result from the use or the inability to use RACDK.
- The RACDK package may not be distributed in a modified form with the exception of other archive formats. This applies to adverts which may not be included as either a file, nor as a modification to an existing .DIZ informational file.
- Any BBS/distribution adverts of any kind may only be included in the archive banner.
- RACDK may be included on shareware CD's and floppy disks providing that any charge levied is for the compilation and not expressly for RACDK.
- If you use this program, you will constitute your agreement to this disclaimer/license.

- All rights reserved worldwide.

*In simple terms: It's free, but it is *ours*. Please also include a small credit somewhere in your programs documentation saying something like "RemoteAccess 'C' Developer Toolkit from Envy Technologies utilised in the development of this product".*

Exclusions

The following parts of RACDK are neither copyright Envy Technologies nor covered by this licence agreement.

Pascal <> C string conversion macros

Taken from the 'C' Snippets collection - Public Domain.

Revision history

v0.00	Original release of converted structs for RA 2.02. Pascal <> C string conversion macros included.
v0.01	Brand new conversion of RA 2.50 gamma structs. * Untested release*
v0.02	Fixed compilation problems. Added enumeration types as per the Pascal structs. Added Pascal <> 'C' string copy macros. Code tidied up & commented.
v0.03	Added new Write format documentation. Added struct and enum prefixes to allow compilation without a C++ compiler. New function library RACDK.H contains: CRC-32 calculation routine Multitasking timeslice Added examples: Using a CRC Reading RA files

Requirements

To use RACDK you require:

- A 'C' language compiler. RACDK has only been tested with Borland C++ v4.52
- A copy of RemoteAccess 2.01, 2.02 or 2.50 Shareware, Hobby or Professional configured & working is recommended.
- Experience with 'C' programming.

Note: The 'C' source files contained within this package have been developed using Borland C++ v4.52 only. Other compilers may interpret this code in such a way that it no longer functions correctly (Common problems include different compilers ideas about longs, ints etc).

Setting up RA-Monitor

To setup RACDK simply unpack the files to a directory. You may want to put the .H files into your 'C' compilers include directory, or add the RACDK path to your compiler's list of include directories.

Using RACDK

Structures

The RA250C.H file included with the package is the core of the RACDK. It defines, in 'C', the structure of the RemoteAccess data & control files so that you may read and write information to work with the RemoteAccess software. This file is based on information in the Pascal structures file included with all releases of RemoteAccess.

There are many files described in the structures document, a listing of them appears in the following section *Overview of files*. A simple example of accessing a RemoteAccess file in 'C' using the RACDK can be found in the included FILEDEMO.C file.

Functions

The RACDK.H file contains functions which you may find of some use while developing applications to work with RemoteAccess. You may either include this file in your project with `#include` or manually copy and paste the routines into your own code. The functions are detailed below however additional detailed information can be found at the start of RACDK.H.

CRC32 - Calculate 32-bit CRC

Prototype: unsigned long CRC32(unsigned char *pBuffer,unsigned int iBufLen)
Input: *pBufferPointer to area of memory to calculate CRC from
iBufLen Length of pBuffer (number of bytes to process)
Output: 32-bit CRC code for memory block
Description: This routine generates a 32-bit CRC code from a block of data. You can pass this routine a block of text such as a password and it will generate the same CRC code as RemoteAccess does when changing a password. This is also used to encode user names for fast lookup as well as in the JAM message base and on Z-Modem transfer packets.
Example: USERrec.PasswordCRC = CRC32(myPass,strlen(myPass));
This would store the password contained in the string myPass in a USER.BBS record ready to be written back to USERS.BBS itself.
Note: This routine uses 'C' string types, to use it on Pascal types, see the section titled *Pascal string handling* later in this document.

TimeSlice - Release timeslice to multitasking OS

Prototype: void TimeSlice(void)
Input: None
Output: None
Description: This routine will attempt to detect DESQview, OS/2 or Windows multitasking operating systems and will release a timeslice to each of them when called. If none are found this routine will release timeslices using the generic method of executing assembler NUL instructions.
Example: while(!kbhit()) TimeSlice();
This would wait for a key to be pressed but will take minimal CPU usage while waiting.
Note: This function will compile only in compilers which support inline assembler.
This function will run only in DOS applications with Intel 80x86 support.

Pascal string handling

Pascal handles strings differently to 'C'. In 'C' a string consists of a sequence of characters with the null character (00h) telling 'C' when the string finishes. This has the advantage of allowing unlimited length strings at the expensive of not being able to tell how long the string is until you have read every character in it (this is what the strlen() function does).

Pascal strings are stored differently. Instead of being null-terminated the first actual byte contains the length of the string, followed by the string itself. For example:-

'C'	unsigned char sUserName[61]; strcpy(sFileName,"Guard"); <i>In memory (hex):</i> 477561726400
Pascal	sUserName: String[5]; sUserName := 'Guard'; <i>In memory(hex):</i> 054775617264

It is important to note that as Pascal stores this length that you do not read more characters than specified as often Pascal will overwrite only part of a string. For example if a Pascal field contained the string 'This is a test' and then later contained the string 'What' if you went and uses the entire contents of the field instead of the first 4 characters specified you would get 'What is a test'.

To help relieve this problem the RACDK contains four macros for managing Pascal <> 'C' string conversions. These are implemented as macros for optimum speed but you are free to convert them to functions (but do not redistribute these functions with the RACDK).

The macros are as follows:

Pas2C	Convert Pascal string to 'C' format <i>Example:</i> <i>Pas2C(CONFIGrec.SysopName);</i> <i>printf("Sysop is %s\n",CONFIGrec.SysopName);</i>
C2Pas	Convert 'C' string to Pascal format <i>Example:</i> <i>C2Pas(CONFIGrec.SysopName);</i> <i>fwrite...</i>
cpyPas2C	Copy a Pascal string into a 'C' string <i>Example:</i> <i>cpyPas2C(CONFIGrec.SysopName,sSysopName);</i> <i>printf("Sysop is %s\n",sSysopName);</i>
cpyC2Pas	Copy a 'C' string to a Pascal string <i>Example:</i> <i>cpyC2Pas(sSysopName,CONFIGrec.SysopName);</i> <i>fwrite...</i>

General tips

Files

- Always open the files in shared mode
- Open a file, read it's contents and close ASAP to prevent RA from being locked-out.
- Some files have multiple records. What this means is that the same information is stored multiple times. For example, the CONFIGrecord found in CONFIG.RA is only stored once - the CONFIGrecord described all the fields contained within this file and matches it exactly. However, the FILESrecord found in FILES.RA is stored many many times, one for each file area. You can choose which area you read using the 'C' function fseek before reading. fseek is based on moving the files current position by a number of bytes so to get to a specific record you calculate the position as (recordnumber-1)*sizeof(record).

Interface

- Making your program appear to the sysops using an RA style interface is a great way to make them feel immediately at home. If your application does uses this interface then you should include a line stating 'Original interface design by Joaquim Homrighausen' somewhere in the documentation.
- Always release timeslices when possible, especially when waiting for a key-press.

Appendix

Frequently asked questions

- Q: The configuration files are reading in garbage, why?
A: Ensure you are loading the correct file properly. If you are sure you are loading it correctly then you **MUST** make sure that your compiler has the option for **ENUM as character** and not **ENUM as integer** which is the default on some 'C' compilers including Borland C++.

File summary

This package consists of the following 9 files:

Main files

RACDK.WRI	Windows Write format documentation
RACDK.DOC	Plain DOS text format documentation
RA250C.H	RemoteAccess 2.50 'C' file structures
RACDK.H	RACDK function library

Examples

TEST-CRC.C	Show how to generate an RA CRC value
FILEDEMO.C	Show how to access RA files & information

Administration

FILE_ID.DIZ	Archive description
DESCRIPT.ION	4-DOS extended file descriptions
WHATS.NEW	Information on changes since v0.2

Contacting Envy Technologies

There are several ways of contacting Envy Technologies. They are:

FidoNet:	2:255/78 or 2:255/119 (Damien Guard)
RANet:	73:7440/78 (Damien Guard)
BBS (UK):	(01481) 727140/727141 (28k8)
BBS (Internat.):	+44-1481-727140/727141 (28k8)
Internet:	envy@guernsey.net

To receive the latest versions of Envy Technologies releases:

FidoNet:	FREQ from 2:255/78 or 2:255/119
	RACDK for latest version of RACDK
	ENVY for list of releases to dates + future release info
BBS:	Log onto our BBS (numbers listed above) as user GUEST.
	Go into the file menu, select group [Ice] Distribution then file area Envy Technologies.
Internet WWW:	http://www.guernsey.net/~envy

Future enhancements

Expect some more example programs and a greatly expanded function library (Note sending, checking user online etc) maybe even complete little applications to show various functions. If you want to see a product that has used RACDK, our own FREEWARE RA-Monitor tool uses this package (shows status of all nodes, send notes, log off users) can be FREQ'ed from our FidoNet address as RAMON or retrieved from our WWW site on the Internet. As of writing the latest version is 1.00.

Other sources

There are many other archives which you may find useful when developing applications for RemoteAccess in 'C', a couple of the most useful ones are listed here and should be available from any good BBS (and are available on our BBS once your account has been validated):

Messages

JAM-API Set of 'C' and Pascal routines for manipulating the JAM message base including reading/writing messages. Also includes HMB2JAM message base converter with source code which is useful if you want to know how to read Hudson message bases.

Doors

OpenDoors OpenDoors v6 is an excellent library for 'C'/'C++' which lets you write a BBS door almost as easily as writing a normal DOS 'C' program.

Credits & acknowledgements

Thanks to Steve Streeting for information on pointers (again ;-) and to those RACDK users who have given good feedback!

Development tools: Borland C++ v4.52
Documentation: Windows Write

Thank-you for using RACDK,

/Jamien

Damien Guard - Envy Technologies - Monday 29th April, 1996.