Using Microsoft® Windows® DLLs within SAS® Programs

Rajesh Lal
Sr. SAS Programmer
Business Intelligence and Analytics Practice

Tuesday, March 04, 2014

Experis™
ManpowerGroup

Agenda

- Introduction
- MODULE family of functions
- Searching for required functionality on MSDN
- SASCBTBL
- Converting routine definition to SASCBTBL
- Examples
  - Creating an encrypted file output
  - Creating an output in an unknown environment based upon software installed
  - Simultaneous multi-user write access

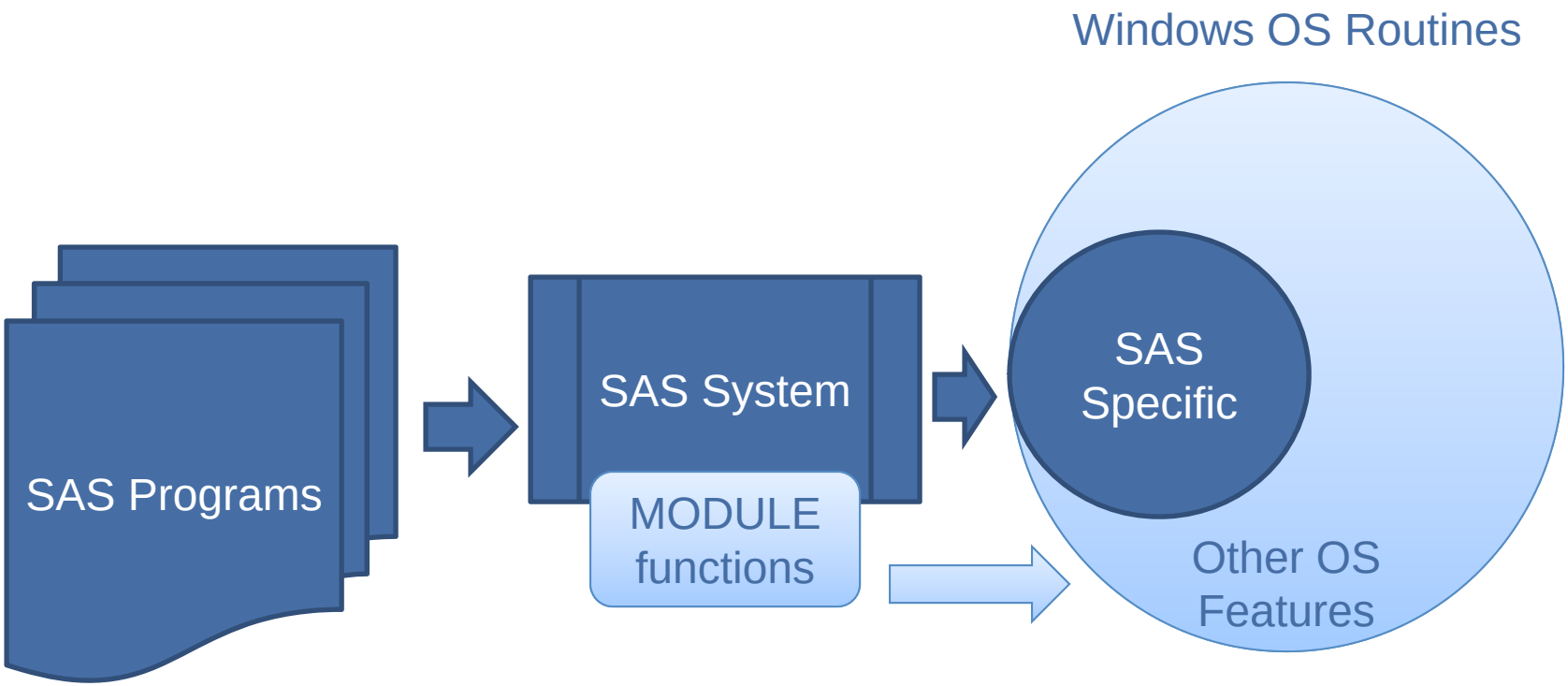
Experis™
ManpowerGroup

Introduction

- DLL: Dynamic Link Library
- Microsoft's implementation of the shared library concept
- Memory Sharing + Dynamic Linking = Optimization
- SAS functions & call routines
  - mathematical, logical, date/time, statistical, financial, character string management and file management categories
  - Provide various OS level functionality
  - Easy to use
- Other OS level functionality
  - No direct functions or call routines
  - MODULE family of functions

Experis™
ManpowerGroup

Overview


Windows OS Routines
SAS Programs
SAS System
MODULE functions
SAS Specific
Other OS Features

MODULE Family of Functions

- CALL MODULE is used to call the routines that do not return any value
- MODULEN is used to call the routines that returns a numeric value
- MODULEC is used to call the routines that returns a character value
- CALL MODULEI, MODULEIN and MODULEIC are used to call the routines that require vector and matrix arguments and return no value, numeric value and character value respectively.

Experis™
ManpowerGroup

Searching on MSDN

- MSDN: Microsoft Developer Network: http://msdn.microsoft.com
- tremendous amount of documentation about the routines available in Windows DLL files
- E.g. Output file encryption – no direct SAS function

MSDN

Windows | Dev Center - Desktop

DASHBOARD GET STARTED DESIGN DEVELOP CERTIFY

Desktop technologies Server and system API index Samples Community

- Desktop App UI
- Desktop Environment
- Application Installation and Servicing
- Audio and Video
- Data Access and Storage
  - Background Intelligent Transfer Service
  - Backup
  - Common Log File System
  - Deployment Image Servicing and Management (DISM) API
  - Distributed File System
  - Distributed File System Replication
  - Extensible Storage Engine
  - File Management API (FMAPI)
  - Image Mastering API
  - Imaging API
  - iSCSI Discovery Library API
  - iSCSI Software Target API
  - Local File Systems
    - Directory Management
    - Disk Management
    - File Management
      - About File Management
      - Using File Management
      - File Management Reference
        - File Management Constants
        - File Management Control Codes
        - File Management Enumerations
        - File Management Functions
        - File Management Structures

Develop desk

24 out of 39 rated this helpful - Rate t

- Application installation and servicing
- Audio and video
- Data access and storage
- Diagnostics
- Documents and printing
- Graphics and gaming
- Networking and Internet
- Security and identity
- System Services
- Desktop App UI
- Desktop environment

De

- Local File Systems
  - Directory Management
  - Disk Management
  - File Management
    - About File Management
    - Using File Management
    - File Management Reference
      - File Management Constants
      - File Management Control Codes
      - File Management Enumerations
      - File Management Functions
      - File Management Structures

EncryptFile function

Encrypts a file or directory. All data streams in a file are encrypted. All new files created in an encrypted directory are encrypted.

Syntax

C++

```
BOOL WINAPI EncryptFile(
  _In_  LPCTSTR lpFileName
);
```

Parameters

lpFileName [in]

The name of the file or directory to be encrypted.

The caller must have the **FILE_READ_DATA**, **FILE_WRITE_DATA**, **FILE_READ_ATTRIBUTES**, **FILE_WRITE_ATTRIBUTES**, and **SYNCHRONIZE** access rights. For more information, see File Security and Access Rights.

Return value

If the function succeeds, the return value is nonzero.

If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

SASCBTBL Attribute Table

- Text file describes the routine to be called
- Contains a description for each DLL routine we intend to call
- Helps MODULE functions:
  - Interpret
  - Validate
  - Convert supplied arguments
- Must be referenced as SASCBTBL file reference:

```
filename SASCBTBL ‘c:\DLLs\attribtbl.txt’;
```

SASCBTBL from DLL routine definition

MSDN Definition

```
BOOL WINAPI EncryptFile(
        _In_ LPCTSTR lpFileName
);
```

Library	Advapi32.lib
DLL	Advapi32.dll
Unicode and ANSI names	EncryptFileW (Unicode) and EncryptFileA (ANSI)

SASCBTBL

```
routine EncryptFileA
  module = advapi32
  minarg = 1
  maxarg = 1
  stackpop = called
  returns  = long
;
arg 1 input char format=$cstr200.;  * lpFileName;
```

Experis
ManpowerGroup

SASCBTBL from DLL routine definition

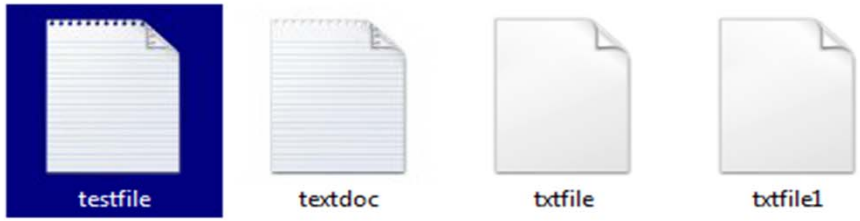
C Language Data Type	SAS Format/Informat
Double	RB8.
Float	FLOAT4.
signed int	IB4.
signed short	IB2.
signed long	IB4.
char *	IB4. (32 bit SAS)
char *	IB8 (x64 and Itanium SAS)
unsigned int	PIB4.
unsigned short	PIB2.
unsigned long	PIB4.
char[w]	$CHARw. or $CSTRw.

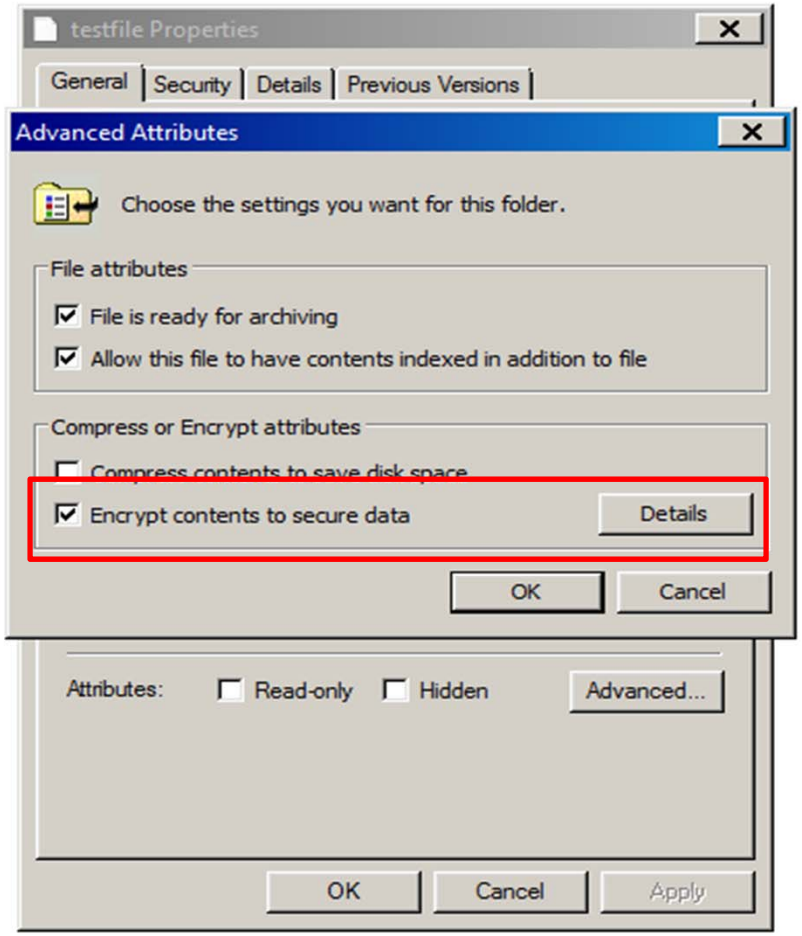
Creating an encrypted file output

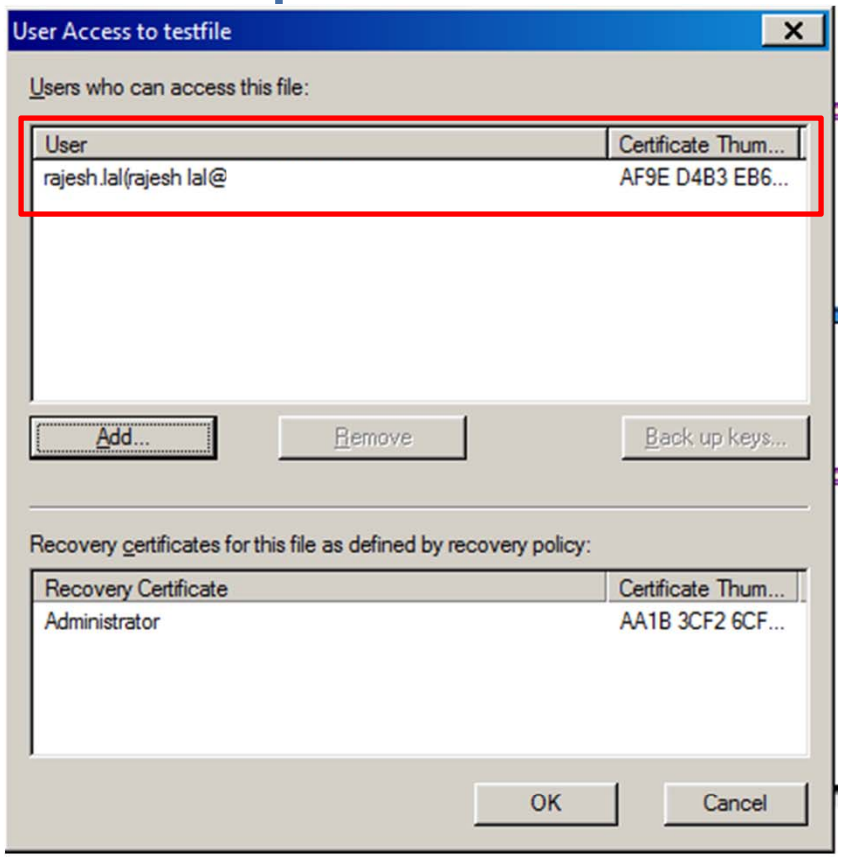
```
filename SASCBTBL 'c:\DLLs\attribtbl.txt';
 DATA _NULL_;
  From_file="C:\Users\rajesh.lal\Desktop\test\textfile.txt";
  Rc1=modulen(‘*e’,‘EncryptFileA’,From_file);
  Put rc1= ;
Run;
```

- Cryptographic method protection of individual files
- NTFS file system
- Uses public-key system

Creating an encrypted file output







- AddUsersToEncryptedFile
- DecryptFileA

Experis™
ManpowerGroup

Creating an output in an unknown environment based upon software installed

- XLS or XLSX
- DOC or DOCX or RTF or TXT
- HKEY_LOCAL_MACHINE\Software\Microsoft\Office\12.0\Common\InstallRoot::Path

MS Office Version	Registry Key Value
Office 2000	9.0
Office XP	10.0
Office 2003	11.0
Office 2007	12.0
Office 2010	14.0

Creating an output in an unknown environment based upon software installed

MSDN Definition

```
LONG WINAPI RegOpenKeyEx(
  _In_        HKEY      hKey,
  _In_opt_    LPCTSTR   lpSubKey,
  _Reserved_  DWORD     ulOptions,
  _In_        REGSAM    samDesired,
  _Out_       PHKEY     phkResult
);
```

Library	Advapi32.lib
DLL	Advapi32.dll
Unicode and ANSI names	EncryptFileW (Unicode) and EncryptFileA (ANSI)

SASCBTBL

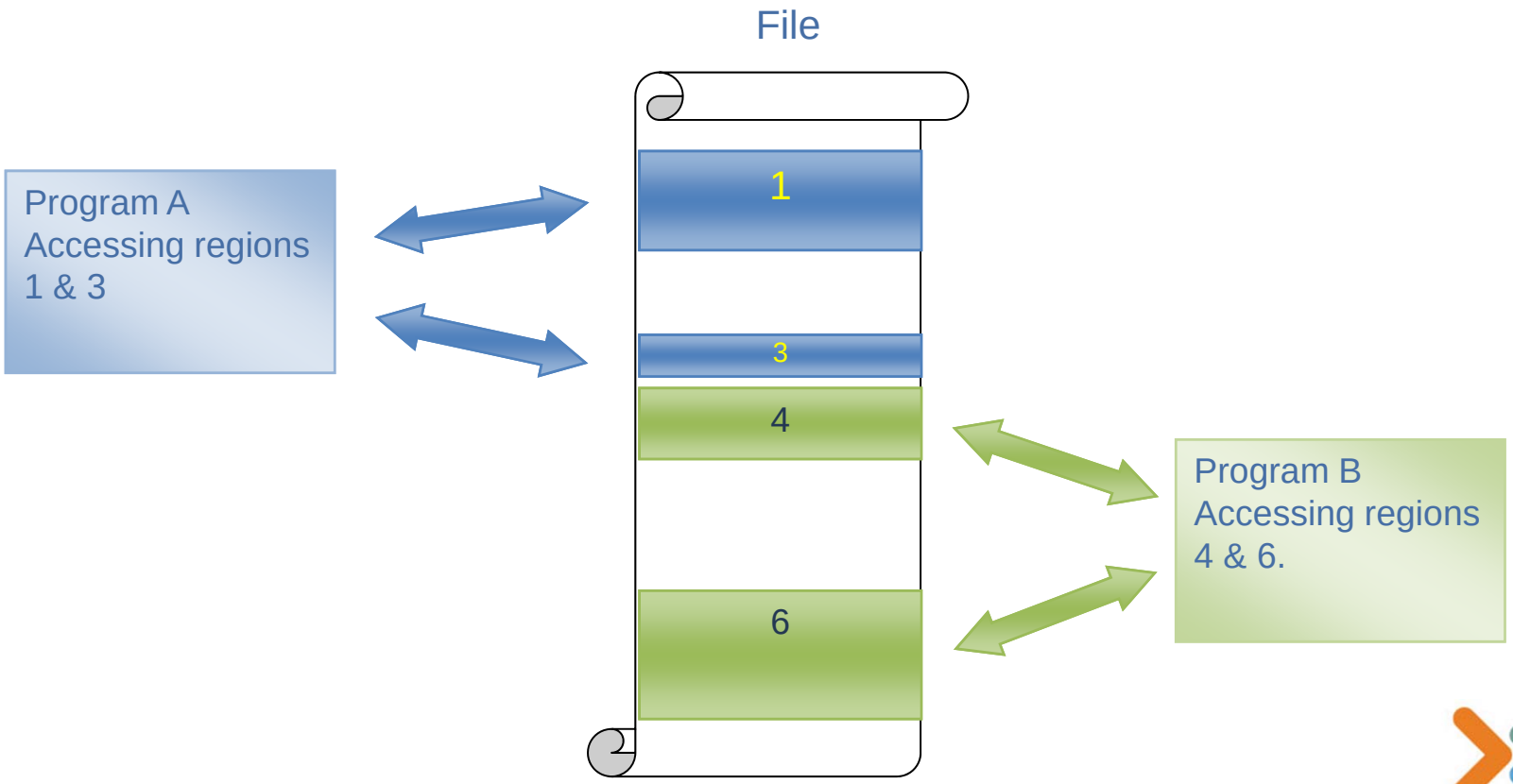
```
routine RegOpenKeyExA
  module = advapi32   /* MSDN defines the DLL name */
  minarg = 5
  maxarg = 5
  stackpop = called
  returns  = long;   /* Return value type */
arg 1 input  num  byvalue format = pib4.;
arg 2 input  char byaddr  format = $cstr200.;
arg 3 input  num  byvalue format = pib4.;
arg 4 input  num  byvalue format = pib4.;
arg 5 update num  byaddr  format = pib4.;
```

Creating an output in an unknown environment based upon software installed

```
%let HKEY_LOCAL_MACHINE    = 80000002x;
%let KEY_QUERY_VALUE = 00000001x;
Node='Software\Microsoft\Office\12.0\Common\InstallRoot';
rc = ModuleN ( '*e'
        , 'RegOpenKeyExA'
        , &HKEY_LOCAL_MACHINE
        , node
        , 0
        , &KEY_QUERY_VALUE
        , hkey1 );
```

MSDN defined constants

Simultaneous multi-user write access



Simultaneous multi-user write access

```
DATA _NULL_;
  From_file="C:\Users\rajesh.lal\Desktop\test\txtfile1";
  GENERIC_ACCESS = 10000000x; *** value for GENERIC_ALL access type constant ;
  FILE_SHARE = 1;
  OPEN_EXISTING = 3;
*** open the file and get the handle;
  hFile = modulen
   ('CreateFileA',From_file,GENERIC_ACCESS,0,0,OPEN_EXISTING,0,0);
*** lock the region from byte 10 to 100;
  rc1=modulen ('LockFile', hfile, 0,00000002x, 10, 100);
*** write the file;
rc1=modulen ('WriteFile', hfile, Buffer, 90);
```

Simultaneous multi-user write access

```
*** unlock the region from byte 10 to 100;
  rc2=modulen ('UnlockFile',hfile,0,0,10,100);
*** close the file;
  rc3=modulen ('CloseHandle', hFile);
  Put hFile= rc1= rc2= rc3=;
run;
```

Conclusion

- using the MODULE family of functions
- searching for required functionality on MSDN
- converting the routine definition to SASCBTBL attribute table
- access the wealth of Operating System functionality
- SAS programs or applications can be designed to be more
  - powerful,
  - dynamic,
  - efficient.

Experis™
ManpowerGroup

Thank You

Rajesh Lal
Sr. SAS Programmer

Experis Business Intelligence and Analytics
5220 Lovers Lane
Portage, MI 49002

Work: (269) 553-5147
Rajesh.Lal@experis.com