

Somar™ RegEdit™ Version 1.2 Help Contents

Copyright © 1995 Somar Software

Send problem reports and comments to 72202.2574@compuserve.com.

For information about other Somar products, visit the Somar Software Web site at <http://www.somar.com>.

[Overview](#)

[DLL Functions](#)

[Example of usage](#)

[VBA Declarations](#)

[Copyright/License/Warranty Disclaimer](#)

[Registration](#)

Overview

Somar RegEdit is a DLL that can be called from 32 bit Visual Basic programs. It allows a network administrator to write a short Visual Basic program to dump or modify the Windows NT user profiles for a large number of users at once.

Example of usage shows how to use a short VBA for Excel program to change the Microsoft Mail server path for all users at once.

In order to use Somar RegEdit effectively, you need to assign user profiles to all users. You can do this for all users at once in User Manager by selecting all users, then specifying a string like `\\SERVER1\PROFILES\%USERNAME%.USR` for the profile path. In order to get a formatted list of current user profiles, in format suitable for copying into an Excel or other spreadsheet, use Somar DumpAcl (available as DMPACL.ZIP in the WINSHARE forum on CompuServe, or at CICA and other Internet sites, including the low-bandwidth site <ftp://somar.com>).

Some notes about profiles:

1) If a user does not currently have an assigned profile, and you assign one using User Manager, then the next time the user logs on, an error message will appear on the user's workstation stating that the user profile cannot be found and the cached profile will be used instead. The cached profile is the copy of the profile on the user's workstation in `%SYSTEMROOT%\CONFIG` that contains the profile values the user has been using in the past. When the user logs off, this cached profile will be copied automatically to the profile path. Thereafter, everything will work normally.

2) Users must have write access to their profile. If the user does not currently have a profile in the profile directory, then the user needs write access to this directory. Ordinary users never need read or write access to the other user profiles in the profile directory. Users who will be using Somar RegEdit will need write access to any profiles they will be modifying. Typically, the profile directory permissions should be set to give add permission to all users, and full control to CREATOR OWNER, Administrators and SYSTEM.

3) When a user logs on, NT compares the timestamps between the cached copy of the profile in `%SYSTEMROOT%\CONFIG` on the user's workstation and the user profile on the server. The most recently changed of these profiles is used. In the case where the server profile has been modified using Somar RegEdit, the server profile will be more recent and so is used.

4) If Somar RegEdit or registry editor is used to edit the copy of the users profile stored on the server, then there are 5 possibilities:

User not logged on during edit session (likely)

When user next logs on, profile will be copied from server to workstation.

User logged on during entire edit session (likely)

Workstation profile will overwrite copy on server, causing edits to be lost. Same effect as if edits never took place.

User logged on at start of edit session, logs off while edit in progress (unlikely)

Workstation profile will not be copied to server, due to file in use on server. Any updates to profile made by user during this logon session will be lost. Similar to case where server online when user logs on, but offline when user logs off.

User not logged at start of edit session, logs on during edit session, remains logged on (unlikely)

User will be unable to obtain copy of profile from server during logon, due to file in use on server. Nasty message to user. User will use locally cached or default profile. When user logs off,

workstation profile (cached or default) will be written to server, overwriting any RegEdit edits.
Similar to case where server is offline when user logs on, and online when user logs off.

User not logged at start of edit session, logs on and off during edit session (very unlikely)
User will be unable to obtain copy of profile from server during logon, due to file in use on server.
Nasty message to user. Workstation profile will not be copied to server, due to file in use on server. Similar to case where server is offline when user logs on and when user logs off.

DLL Functions

Note on error handling

All functions return a status code, which is zero for success and non-zero for failure. In cases of severe error, RGEDIT.DLL displays a message box with diagnostic information. Here is a pseudo-code example of handling errors:

```
Do for all users
    rc = RegLoadProfile(...)
    If rc = 0 Then
        rc = RegSetValue(...)
        rc = RegUnloadProfile(...)
    Endif
Loop
```

Note how the program does not check the return code from RegSetValue or RegUnloadProfile. This is because RGEDIT.DLL will already have displayed a message box and there is nothing more the program can do about failures in these functions. On the other hand, the return code from RegLoadProfile is checked, to avoid getting more message boxes if RegSetValue or RegUnloadProfile is called with a bad hProfile parameter.

Note on registry paths

Many of the functions take a RegPath parameter. Components of the registry path are separated by "\". The initial components compose a parent key path. In the case of functions which operate on registry keys, the last component is a key name. In the case of functions which operate on registry values, the last component is a value name. If a key is empty, then it indicates the entire profile. For example:

```
"Console\Command Prompt"
    "Console" is the parent key path
    "Command Prompt" is the key affected by the function

"Console\Command Prompt\WindowSize"
    "Console\Command Prompt" is the parent key path
    "WindowSize" is the value affected by the function

"Console"
    "" is the parent key path (root of profile)
    "Console" is the key affected by the function
""
    "" is the parent key path (root of profile)
    "" is the key affected by the function (entire profile)
```

RgedLoadProfile RgedCreateProfile

```
Declare Function RgedLoadProfile Lib "rgedit.dll" _
    (ByVal ProfileName As String, _
    ByVal ProfilePath As String, _
    hProfile As Long) As Long

Declare Function RgedCreateProfile Lib "rgedit.dll" _
    (ByVal ProfileName As String, _
    ByVal ProfilePath As String, _
    hProfile As Long) As Long
```

Example:

```
rc = RgedLoadProfile("USER1", "C:\PROFILES\USER1.USR", hProfile)
rc = RgedCreateProfile("USER1", "C:\PROFILES\USER1.USR", hProfile)
```

Notes:

- 1) This function is similar to the Load Hive command in Registry Editor.
- 2) The ProfileName is typically the UserName. This is the name of the new key that will be created under HKEY_USERS.
- 3) For RgedLoadProfile, the ProfilePath must exist, or a message box will be displayed. For RgedCreateProfile, the ProfilePath file will be created if it does not exist, else the existing file will be used.
- 3) You must have Backup and Restore from Backup rights to load profiles. Members of the Administrators and Backup Operators groups have these rights.
- 4) RGEDIT.DLL will display a message box and return a non-zero return code if the load of the profile fails.

RgedUnloadProfile

```
Declare Function RgedUnloadProfile Lib "rgedit.dll" _
    (ByVal hProfile As Long) As Long
```

Example:

```
rc = RgedUnloadProfile (hProfile)
```

Notes:

- 1) This function is similar to the Unload Hive menu command in Registry Editor. If for some reason this function is not called (for example, you abort processing) then you can use the Unload Hive menu command to do what this function does.
- 2) RGEDIT.DLL will display a message box and return a non-zero return code if unload fails.

RgedOpenCurrentUser

```
Declare Function RgedOpenCurrentUser Lib "rgedit.dll" _
    (hProfile As Long) As Long
```

Example:

```
rc = RgedOpenCurrentUser(hProfile)
```

Notes:

- 1) This function does nothing except set hProfile to access the HKEY_CURRENT_USER registry hive.
- 2) RgedUnloadProfile has no effect for the hProfile returned by this function.
- 3) Return code is always 0.

RgedCopyKey

```
Declare Function RgedCopyKey Lib "rgedit.dll" _
    (ByVal hProfileFrom As Long, _
    ByVal hProfileTo As Long, _
    ByVal RegPath as String) As Long
```

Example:

```
rc = RgedCopyKey (hProfile1, hProfile2, "Control Panel\Desktop")
```

Notes:

- 1) The specified key and all its subkeys and values are copied.
- 2) The entire key specified by RegPath must exist in the FROM profile. Only the parent key part of RegPath must exist in the TO profile.
- 3) Items (keys and values) in both the FROM and TO profiles are replaced.
- 4) Items in only the TO profile are left untouched.
- 5) Items in only the FROM profile are added.
- 6) RGEDIT.DLL will display a message box and return a non-zero return code if unable to copy all keys and values successfully.

RgedCopyValue

```
Declare Function RgedCopyValue Lib "rgedit.dll" _
    (ByVal hProfileFrom As Long, _
    ByVal hProfileTo As Long, _
    ByVal RegPath As String) As Long
```

Example:

```
rc = RgedCopyValue (hProfile1, hProfile2, "Console\WindowSize")
```

Notes:

- 1) The key part of RegPath must exist in both FROM and TO profiles.
- 2) The value specified by RegPath must exist in the FROM profile.
- 4) If the value specified by RegPath exists in the TO profile it is replaced, else it is added.
- 5) RGEDIT.DLL will display a message box and return a non-zero return code if copy fails.

RgedCreateKey

```
Declare Function RgedCreateKey Lib "rgedit.dll" _
    (ByVal hProfile As Long, _
    ByVal RegPath As String) As Long
```

Example:

```
rc = RgedCreateKey (hProfile, "Console\New Key")
```

Notes:

- 1) The parent key part of RegPath must exist. The new key specified by RegPath will be created if it does not exist. For example, if RegPath is "Console\New Key", then "Console" must already exist and "New Key" will be created if it doesn't already exist, with no error if it does exist.
- 2) RGEDIT.DLL will display a message box and return a non-zero return code if the create fails.

RgedTestKey

```
Declare Function RgedTestKey Lib "rgedit.dll" _
    (ByVal hProfile As Long, _
    ByVal RegPath As String) As Long
```

Example:

```
rc = RgedTestKey (hProfile, "Console\New Key")
```

Notes:

- 1) Return code is zero if the RegPath key exists, otherwise non-zero.

RgedDeleteKey

```
Declare Function RgedDeleteKey Lib "rgedit.dll" _  
  (ByVal hProfile As Long, _  
   ByVal RegPath As String) As Long
```

Example:

```
rc = RgedDeleteKey (hProfile, "Console\New Key")
```

Notes:

- 1) The parent key part of RegPath must exist. The key specified by RegPath and all its subkeys and values will be deleted, if they exist. For example, if RegPath is "Console\New Key", then "Console" must exist and "New Key" and its subkeys and values will be deleted if it exists.
- 2) RGEDIT.DLL will display a message box and return a non-zero return code if the delete fails, except for the case where the key to be deleted does not exist, in which case only a non-zero return code is returned.

RgedDeleteValue

```
Declare Function RgedDeleteValue Lib "rgedit.dll" _  
  (ByVal hProfile As Long, _  
   ByVal RegPath As String) As Long
```

Example:

```
rc = RgedDeleteValue (hProfile, "Console\New Key\WindowSize")
```

Notes:

- 1) The key part of RegPath must exist. The value specified by RegPath will be deleted if it exists.
- 2) RGEDIT.DLL will display a message box and return a non-zero return code if the delete fails, except for the case where the value to be deleted does not exist, in which case only a non-zero return code is returned.

RgedGetLastModifiedTime

```
Declare Function RgedGetLastModifiedTime Lib "rgedit.dll" _  
  (ByVal hProfile As Long, _  
   ByVal RegPath As String, _  
   wYear As Integer, wMonth As Integer, wDay As Integer, _  
   wHour As Integer, wMinute As Integer, wSecond As Integer) As Long
```

Example:

```
rc = RgedGetLastModifiedTime (hProfile, "Console")
```

Notes:

- 1) RegPath must specify a key. There is no way to get the last modified time for a value. The last modified time is updated for the key whenever any of the values of the key are modified.
- 2) wYear is the full year (e.g. 1995), wMonth is 1 to 12, wDay is 1 to 31, wHour is 0 to 23, wMinute and wSecond are 0 to 59.

- 3) REGEDIT.DLL will display a message box and return a non-zero return code if a severe error occurs. No message box is displayed but a non-zero return code is returned if the key specified by RegPath does not exist.

RgedEnumKeysInit
RgedEnumKeys
RgedEnumValuesInit
RgedEnumValues
RgedEnumCancel

```
Declare Function RgedEnumKeysInit Lib "rgedit.dll" _  
  (ByVal hProfile As Long, _  
   ByVal RegPath As String, _  
   hEnum As Long) As Long
```

```
Declare Function RgedEnumValuesInit Lib "rgedit.dll" _  
  (ByVal hProfile As Long, _  
   ByVal RegPath As String, _  
   hEnum As Long) As Long
```

```
Declare Function RgedEnumKeys Lib "rgedit.dll" _  
  (ByVal hEnum As Long, _  
   ByVal NameBuf As String, _  
   ByVal NameMax As Long, _  
   NameLen Long) As Long
```

```
Declare Function RgedEnumValues Lib "rgedit.dll" _  
  (ByVal hEnum As Long, _  
   ByVal NameBuf As String, _  
   ByVal NameMax As Long, _  
   NameLen As Long, _  
   ValueType As Long) As Long
```

```
Declare Function RgedEnumCancel Lib "rgedit.dll" _  
  (ByVal hEnum As Long) As Long
```

Examples:

```
Const REG_SZ = 1  
Const REG_BINARY = 3  
Const REG_DWORD = 4  
Dim s As String, t As Long, i As Long, hEnum as Long, v As String
```

```
rc = RgedEnumKeysInit (hProfile, "Console", hEnum)  
Do While (rc = 0)  
  s = String$(1000, " ")  
  rc = RgedEnumKeys (hEnum, s, 1000, i)  
  If rc <> 0 Then Exit Do  
  s = Left$(s, i)  
  If s = "Command Prompt" Then  
    rc = RgedEnumCancel (hEnum)  
    Exit Do  
  Endif  
Loop
```

```
rc = RgedEnumValuesInit (hProfile, "Console", hEnum)  
Do While (rc = 0)  
  s = String$(1000, " ")
```



```

rc = RgedEnumValues (hEnum, s, 1000, i, t)
If rc <> 0 Then Exit Do
s = "Console\" + Left$(s, i)
if t = REG_SZ
    v = String$(1000, " ")
    RgedGetString (hProfile, s, v, 1000, i)
Elseif t = REG_DWORD
    RgedGetDword (hProfile, s, i)
Else
    v = String$(1000, "0")
    RgedGetValue (hProfile, s, ByVal v, 1000, i, t)
Endif
Loop

```

Notes:

- 1) hEnum points to an allocated structure in RGEDIT.DLL that is automatically deallocated by RGEDIT.DLL when RgedEnumKeys or RgedEnumValues returns a non-zero return code. This structure can also be freed by calling RgedEnumCancel. If the structure is not freed, then the unload of the profile will fail.
- 2) RgedEnumKeys and RgedEnumValues blank fill NameBuf so that TRIM\$ will work. There is no terminating null.
- 3) See RgedGetValue for a description of the values for the ValueType parameter.
- 4) RGEDIT.DLL will display a message box and return a non-zero return code if a severe error occurs during enumeration. No message box is displayed but a non-zero return code is returned from RgedEnumKeysInit and RgedEnumValuesInit if the key specified by RegPath does not exist. A non-zero return code from RgedEnumKeys and RgedEnumValues is normal, and indicates that there are no more items to enumerate.

RgedGetString

RgedGetDword

RgedGetValue

```

Declare Function RgedGetString Lib "rgedit.dll" _
    (ByVal hProfile As Long, _
    ByVal RegPath As String, _
    ByVal DataBuf As Any, _
    ByVal BufSize As Long, _
    ByVal BufLen As Long) As Long

```

```

Declare Function RgedGetDword Lib "rgedit.dll" _
    (ByVal hProfile As Long, _
    ByVal RegPath As String, _
    DataBuf As Long) As Long

```

```

Declare Function RgedGetValue Lib "rgedit.dll" _
    (ByVal hProfile As Long, _
    ByVal RegPath As String, _
    DataBuf As Any, _
    ByVal BufSize As Long, _
    ByVal BufLen As Long, _
    ValueType As Long) As Long

```

Examples:

```
Dim s As String, t As Long, i As Long

s = String$(1000, " ")
rc = RgedGetString (hProfile, "Environment\Temp", s, i)
s = Left$(s, i)

s = String$(1000, " ")
rc = RgedGetValue (hProfile, "Environment\Temp", s, 1000, i, t)
s = Left$(s, i - 1) ' i includes terminating null

rc = RgedGetDword (hProfile, "Console\WindowSize", i)
```

Notes:

- 1) Values for the ValueType output parameter are as follows (see WINNT.H for a complete list):
REG_SZ = 1
REG_BINARY = 3
REG_DWORD = 4
REG_MULTI_SZ = 7
- 2) RegGetString blank fills DataBuf so that TRIM\$ will work. There is no terminating null.
- 3) RGEDIT.DLL will display a message box and return a non-zero return code if the key part of the RegPath does not exist or if the get fails (other than because the value does not exist). If the value name part of RegPath does not exist, then a non-zero return code is returned but there is no message box.

RgedSetString **RgedSetDword** **RgedSetValue**

```
Declare Function RgedSetString Lib "rgedit.dll" _
  (ByVal hProfile As Long, _
  ByVal RegPath As String, _
  ByVal DataBuf As String) As Long
```

```
Declare Function RgedSetDword Lib "rgedit.dll" _
  (ByVal hProfile As Long, _
  ByVal RegPath As String, _
  ByVal DataBuf As Long) As Long
```

```
Declare Function RgedSetValue Lib "rgedit.dll" _
  (ByVal hProfile As Long, _
  ByVal RegPath As String, _
  DataBuf As Any, _
  ByVal BufLen As Long, _
  ByVal ValueType As Long) As Long
```

Example:

```
Global Const REG_SZ = 1
Dim s As String

rc = RgedSetDword (hProfile, "Key\SubKey\Value, 25)

s = "This is some data"
```

```
rc = RgedSetString (hProfile, "Key\SubKey\Value", s)

' add 1 to len(s) for terminating null
rc = RgedSetValue (hProfile, "Key\SubKey\Value", _
    ByVal s, len(s)+1, REG_SZ)
```

Notes:

- 1) If the value name part of RegPath does not exist, it will be created (this is not an error). The key part of RegPath must exist.
- 2) See RegGetValue for a description of the values for the ValueType parameter.
- 3) RGEDIT.DLL will display a message box and return a non-zero return code if the value cannot be set.

Example of Usage

This example shows how to use RGEDIT.DLL together with 32 bit Excel VBA to change the value of the Microsoft mail server path in all user profiles.

```
Dim hUser as Long, row as Long, rc As Long
Dim szUserName as String, szProfilePath as String

For row = 1 to 50
    szUserName = Worksheets("Sheet1").Cells(row, 1)
    szProfilePath = Worksheets("Sheet1").Cells(row, 2)
    rc = RgedLoadProfile(szUser, szProfilePath, hUser)
    If rc = 0 Then
        rc = RgedSetString (hUser, _
            "Software\Microsoft\Mail\Microsoft Mail\ServerPath", _
            "\\server1\common\hidden\wgpo")
        rc = RgedUnloadProfile (hUser)
    Endif
Next
```

VBA Declarations

The following declarations are suitable for copying directly into a VBA program.

```
Declare Function RgedLoadProfile Lib "rgedit.dll" (ByVal ProfileName As String, _  
    ByVal ProfilePath As String, hProfile As Long) As Long  
  
Declare Function RgedCreateProfile Lib "rgedit.dll" (ByVal ProfileName As String, _  
    ByVal ProfilePath As String, hProfile As Long) As Long  
  
Declare Function RgedUnloadProfile Lib "rgedit.dll" (ByVal hProfile As Long) As Long  
  
Declare Function RgedOpenCurrentUser Lib "rgedit.dll" (hProfile As Long) As Long  
  
Declare Function RgedCopyKey Lib "rgedit.dll" (ByVal hProfileFrom As Long, _  
    ByVal hProfileTo As Long, ByVal RegPath As String) As Long  
  
Declare Function RgedCopyValue Lib "rgedit.dll" (ByVal hProfileFrom As Long, _  
    ByVal hProfileTo As Long, ByVal RegPath As String) As Long  
  
Declare Function RgedCreateKey Lib "rgedit.dll" (ByVal hProfile As Long, _  
    ByVal RegPath As String) As Long  
  
Declare Function RgedTestKey Lib "rgedit.dll" (ByVal hProfile As Long, _  
    ByVal RegPath As String) As Long  
  
Declare Function RgedDeleteKey Lib "rgedit.dll" (ByVal hProfile As Long, _  
    ByVal RegPath As String) As Long  
  
Declare Function RgedDeleteValue Lib "rgedit.dll" (ByVal hProfile As Long, _  
    ByVal RegPath As String) As Long  
  
Declare Function RgedGetLastModifiedTime Lib "rgedit.dll" (ByVal hProfile As Long, _  
    ByVal RegPath As String, wYear As Integer, wMonth As Integer, wDay As Integer, _  
    wHour As Integer, wMinute As Integer, wSecond As Integer) As Long  
  
Declare Function RgedEnumKeysInit Lib "rgedit.dll" (ByVal hProfile As Long, _  
    ByVal RegPath As String, hEnum As Long) As Long  
  
Declare Function RgedEnumKeys Lib "rgedit.dll" (ByVal hEnum As Long, _  
    ByVal NameBuf As String, ByVal NameMax As Long, NameLen As Long) As Long  
  
Declare Function RgedEnumValuesInit Lib "rgedit.dll" (ByVal hProfile As Long, _  
    ByVal RegPath As String, hEnum As Long) As Long  
  
Declare Function RgedEnumValues Lib "rgedit.dll" (ByVal hEnum As Long, _  
    ByVal NameBuf As String, ByVal NameMax As Long, NameLen As Long, _  
    ValueType As Long) As Long
```

```

Declare Function RgedEnumCancel Lib "rgedit.dll" (ByVal hEnum As Long) As Long

Declare Function RgedGetString Lib "rgedit.dll" (ByVal hProfile As Long, _
    ByVal RegPath As String, ByVal DataBuf As String, _
    ByVal BufSize As Long, BufLen As Long) As Long

Declare Function RgedSetString Lib "rgedit.dll" (ByVal hProfile As Long, _
    ByVal RegPath As String, ByVal DataBuf As String) As Long

Declare Function RgedGetDword Lib "rgedit.dll" (ByVal hProfile As Long, _
    ByVal RegPath As String, DataBuf As Long) As Long

Declare Function RgedSetDword Lib "rgedit.dll" (ByVal hProfile As Long, _
    ByVal RegPath As String, ByVal DataBuf As Long) As Long

Declare Function RgedGetValue Lib "rgedit.dll" (ByVal hProfile As Long, _
    ByVal RegPath As String, DataBuf As Any, ByVal BufSize As Long, _
    BufLen As Long, ValueType As Long) As Long

Declare Function RgedSetValue Lib "rgedit.dll" (ByVal hProfile As Long, _
    ByVal RegPath As String, DataBuf As Any, ByVal BufLen As Long, _
    ByVal ValueType As Long) As Long

'
' ValueType values (see WINNT.H for full list)
'
Global Const REG_SZ = 1
Global Const REG_BINARY = 3
Global Const REG_DWORD = 4
Global Const REG_MULTI_SZ = 7

```

Copyright/License/Warranty Disclaimer

Somar RegEdit is Copyright © 1995 Somar Software, All rights reserved.
Send problem reports and other comments to 72202.2574@compuserve.com.

License Agreement and Warranty Disclaimer

You should carefully read the following terms and conditions before using this software. Use of this software indicates your acceptance of these terms and conditions. If you do not agree with them, do not use the software.

Shareware Version

You are hereby licensed to: use the Shareware Version of the software for a 21 day evaluation period; make as many copies of the Shareware version of this software and documentation as you wish; give exact copies of the original Shareware version to anyone; and distribute the Shareware version of the software and documentation in its unmodified form via electronic means. There is no charge for any of the above.

You are specifically prohibited from charging, or requesting donations, for any such copies, however made; and from distributing the software and/or documentation with other products (commercial or otherwise) without prior written permission, with one exception: Disk Vendors approved by the Association of Shareware Professionals are permitted to redistribute Somar RegEdit, subject to the conditions in this license, without specific written permission.

You are specifically prohibited from copying or redistributing the keyfile (RGEDIT.KEY) that will be sent to you upon receipt of your registration payment. Please note that this keyfile contains your registration number and user or company name in encrypted format, and is therefore different for each registered user.

Unregistered use of Somar RegEdit after the 21-day evaluation period is in violation of United States and International copyright laws.

For how to register, see [Registration](#).

A single registered copy of Somar RegEdit can only be installed on a single computer. Thus, if you have 3 computers on which you want to install Somar RegEdit, you must register and pay for 3 copies of Somar RegEdit.

Governing Law

This agreement shall be governed by the laws of the District of Columbia.

Disclaimer of Warranty

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. BECAUSE OF THE VARYING HARDWARE/SOFTWARE ENVIRONMENTS INTO WHICH SOMAR REGEDIT MAY BE PUT, THERE IS NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE.

GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY PROGRAM BE THOROUGHLY TESTED BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM. ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

Registration

This is not free software. This license allows you to use this software for evaluation purposes without charge for a period of 21 days. If you use this software after the 21 day evaluation period a registration fee of \$10 is required. Payments must be in US dollars drawn on a US bank, payable to Somar Software, and sent to:

Somar Software
1 Scott Circle, NW Suite 816
Washington, DC 20036
USA

You can also register through CompuServe (GO SWREG, ID 5060).

Upon receipt of your registration payment (check or CompuServe notice), you will be sent a key file (RGEDIT.KEY), which is to be placed in the same directory as RGEDIT.DLL. This key file will be your proof of registration. The presence of this key file will eliminate a warning notice dialog box that appears when certain functions are called in the unregistered version. Please note that this keyfile contains your registration number and user or company name in encrypted format, and is therefore different for each registered user.

Somar Software reserves the right to increase the registration fee at any time without notice. If the fee has been increased, and your payment is received after such increase, then Somar Software has the right to require additional payment before accepting your registration and sending you a key file. Once you have paid the then current registration fee, and Somar Software has accepted your fee and sent you a key file, you have the right to use the version of Somar RegEdit that you paid for, for as long as you want, without additional payment. Future versions of Somar RegEdit may require a different key file. You may be required to pay an additional fee to upgrade to such future versions.

A single registered copy of Somar RegEdit can only be installed on a single computer. Thus, if you have 3 computers on which you want to install Somar RegEdit, you must register and pay for 3 copies of Somar RegEdit.

