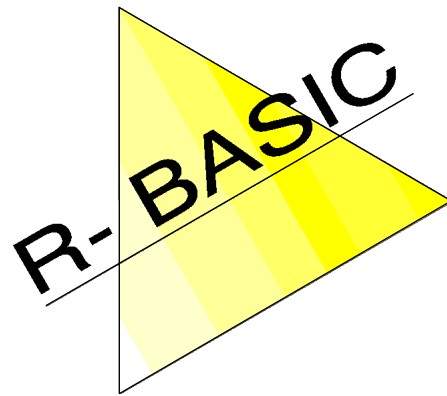


# ***R-BASIC*** ***Extensions***



 GeoRun Library Handbuch

 GeoRun Library Manual

Version 1.0  
Initial Release

## Inhaltsverzeichnis

|   |   |
|---|---|
| 1 Starten und Beenden von anderen Applikationen | 3 |
| 2 Fernsteuern anderer Applikationen             | 8 |

## Table of Contents

|  |    |
|--|----|
| 1 Start and Stop Other Applications    | 13 |
| 2 Remote Control of Other Applications | 17 |

# R-BASIC - Extensions: GeoRun Library

Einfach unter PC/GEOS programmieren

Die GeoRun Library erlaubt dem R-BASIC Programmierer mit anderem GEOS-Applikationen zu kommunizieren. Das können sowohl R-Apps (compilierte R-BASIC Programme) als auch andere GEOS-Applikationen, wie z.B. GeoWrite, sein. Dabei kann bei Bedarf eine Datendatei übergeben werden.

Um die GeoRun Library verwenden zu können, müssen Sie sie includen:

```
Include "GeoRun"
```

## 1 Starten und Beenden von anderen Applikationen

|                        |  |
|------------------------|--|
| GeoRunStartApplication | Übergibt eine Datei an eine Applikation, deren GeodeToken Sie kennen. Bei Bedarf wird die Applikation gestartet. |
| GeoRunOpenFile         | Öffnet eine Datendatei, wobei die zugehörige Applikation aus dem Typ der Datei ermittelt wird.                   |
| GeoRunFindDosApp       | Ermittelt die zu einer DOS-Datei gehörende GEOS Applikation.   |
| GeoRunTestRunning      | Prüft, ob eine bestimmte GEOS-Applikation bereits läuft.   |
| GeoRunCloseAppliation  | Schließt eine laufende GEOS-Applikation.   |

### GeoRunStartApplication

GeoRunStartApplication öffnet eine GEOS-Applikation und übergibt ihr eine Datendatei. Sie müssen selbst darauf achten, dass der Typ der Datei zur Applikation passt.

Sie können GeoRunStartApplication auch verwenden, um einer bereits laufenden Applikation eine (weitere) Datei zu übergeben.

Deklaration:

```
FUNCTION GeoRunStartApplication(appToken as GeodeToken, dataFile as String(32)) as WORD
```

Parameter:

appToken: GeodeToken der zu öffnenden Applikation.

Wenn die Applikation noch nicht läuft, wird sie gestartet. Dazu muss sie sich im World-Verzeichnis, einem seiner Unterverzeichnisse oder im SysAppl-Verzeichnis befinden.

dataFile: Name der zu übergebenden Datei. Die Datei muss sich im aktuellen Ordner befinden und zur zu startenden Applikation passen.  
Ein Leerstring ist zulässig.

## R-BASIC - Extensions: GeoRun Library

Einfach unter PC/GEOS programmieren

**Return:** Null oder Fehlercode. Die globale Variable fileError wird gesetzt. Siehe unten.

**Hinweise:**

- Ist das zugehörige Programm ein R-BASIC Programm, so wird die Datei dem OnStartup-Handler übergeben, falls das Programm neu gestartet werden müsste. Läuft es bereits, so wird die Datei dem OnConnection-Handler übergeben.
- GeoRunStartApplication kann (wie jede Function in R-BASIC) auch wie eine SUB benutzt werden. Siehe Beispiel.

### Details zum Fehlercode

GeoRunStartApplication und GeoRunOpenFile liefern Null, wenn die Applikation erfolgreich gestartet werden konnte bzw. bereits läuft. Andernfalls liefern die Routinen einen Fehlercode. Außerdem setzen diese Routinen die globale Variable fileError (Null oder Fehlercode).

Positive Fehlercodes und Fehlercodes im Bereich von -50 bis Null sind R-BASIC "Standard"-Codes. Sie können die Routine ErrorText\$() verwenden, um eine Beschreibung des Fehlers zu erhalten. Typische Fehlermeldungen sind:

ERROR\_FILE\_NOT\_FOUND: Die Datendatei existiert nicht im aktuellen Ordner

ERROR\_ACCESS\_DENIED: Der Zugriff wurde verweigert. Dieser Fehler tritt auch auf, wenn Sie einen Ordner statt einer Datei angeben.

Die Fehlercode -51 und -52 sind library-interne Fehler. Sie sind folgendermaßen definiert:

```
CONST LIB_ERROR_OPENING_APP      = -51
! Die Applikation konnte nicht gestartet werden, meist weil
! keine Applikation mit dem übergebenen Token gefunden wurde.
CONST LIB_ERROR_NO_DOS_APP      = -52
! Es wurde keine passende DOS-Applikation in der INI gefunden.
! Auch ein *.* - Eintrag existiert nicht.
```

**Beispiel:** Öffnen von GeoWrite mit einer Datei im Document-Ordner

```
DIM AppToken AS GeodeToken

AppToken.tokenchars = "WP00"
AppToken.manufid = 0
SetStandardPath(SP_DOCUMENT)

GeoRunStartApplication(appToken, "Karte Danke")
```

## R-BASIC - Extensions: GeoRun Library

Einfach unter PC/GEOS programmieren

### GeoRunOpenFile

GeoRunOpenFile übergibt eine Datei an die zugehörige Applikation, damit diese sie öffnen kann. Die Datei muss sich im aktuellen Ordner befinden. Für VM-Dateien (z.B. GeoWrite-Dokumente) und GEOS Daten-Dateien wird die zugehörige Applikation aus dem Dateityp ermittelt (Creator-Token), für DOS-Dateien wird die zugehörige Applikation aus der GEOS.INI ( [fileManager] : filenameTokens) gelesen.

Deklaration:

```
FUNCTION GeoRunOpenFile(dataFile as String(32)) as WORD
```

Parameter:

dataFile: Name der zu öffnenden Datei. Die Datei muss sich im aktuellen Ordner befinden. Die zugehörige Applikation wird automatisch ermittelt.

Return: Null oder Fehlercode. Die globale Variable fileError wird gesetzt. Siehe auch: GeoRunStartApplication.

Hinweise:

- Ist das zugehörige Programm ein R-BASIC Programm, so wird die Datei dem OnStartup-Handler übergeben, falls das Programm neu gestartet werden müsste. Läuft es bereits, so wird die Datei dem OnConnection-Handler übergeben.
- GeoRunOpenFile kann (wie jede Function in R-BASIC) auch wie eine SUB benutzt werden.
- Wenn Sie nicht wollen, dass bei einer DOS-Datei, für die keine explizite Zuweisung in der GEOS.INI existiert, die Default-Zurdnung "\*.\*)" verwendet wird, benutzen Sie GeoRunFindDosApp, um zu prüfen, ob eine explizite Zuweisung in der GEOS.INI existiert.
- GeoRunOpenFile kann auch eine Applikation starten, wenn sie sich im aktuellen Ordner befindet.

Beispiel: Öffnen einer Grafik-Datei im Document-Ordner

```
SetStandardPath(SP_DOCUMENT)  
GeoRunOpenFile("MANJUL.GIF")
```

### GeoRunFindDosApp

GeoRunFindDosApp durchsucht die GEOS.INI nach der zu einer DOS-Datei gehörenden Applikation und liefert das zugehörige GeodeToken.

## R-BASIC - Extensions: GeoRun Library

Einfach unter PC/GEOS programmieren

Deklaration:

```
FUNCTION GeoRunFindDosApp(dataFile as String(32), findStar as Integer) as GeodeToken
```

Parameter:

dataFile: Kompletter Name der DOS-Datei. Wenn Sie nur die Dateierweiterung kennen, übergeben Sie hier einen fiktiven Namen, z.B. "X.EXT" oder "\*.EXT".

Übergeben Sie niemals nur die Dateierweiterung! Das wird als Dateiname ohne Erweiterung interpretiert!

findStar: TRUE oder FALSE

TRUE: Falls keine explizite Zuordnung gefunden wurde, wird das Token der "\*" Zuordnung zurückgeliefert - falls eine solche Zuordnung existiert.

FALSE: Falls keine explizite Zuordnung gefunden wurde wird immer eine Null-Token zurückgeliefert. Die Zuordnung zu "\*" wird ignoriert.

Return: GeodeToken der zugehörigen Applikation, oder ein Null-Token, wenn keine zugehörige Applikation gefunden wurde.

Beispiel: Finden des Standard-Browsers (HTML-Betrachter)

```
DIM AppToken AS GeodeToken
  appToken = GeoRunFindDosApp("*.HTM", FALSE)
  MsgBox appToken.tokenChars + "," + Str$(appToken.manufID)
```

Hinweis:

- Um zu prüfen, ob ein Null-Token zurückgeliefert wurde, können Sie die Funktion NullStruct() verwenden.

```
DIM appToken as GeodeToken
  appToken = GeoRunFindDosApp("MYFILE.DTF", FALSE)
  IF appToken = NullStruct() THEN ....
```

### GeoRunTestRunning

GeoRunTestRunning prüft, ob eine bestimmte Applikation bereits läuft, oder nicht.

Deklaration:

```
FUNCTION GeoRunTestRunning(token as GeodeToken) as Integer
```

Parameter:

token: GeodeToken der fraglichen Applikation.

## R-BASIC - Extensions: GeoRun Library

Einfach unter PC/GEOS programmieren

Return: TRUE: Die Applikation läuft bereits.  
FALSE: Die Applikation läuft noch nicht.

**Warnung!** GeoRunTestRunning schließt eine eventuell bestehende Connection zum getesteten Programm. Diese Warnung ist nur relevant, wenn Sie GeoRunOpenConnection (siehe unten) verwenden.

Beispiel: Prüfen, ob GeoWrite schon läuft

```
DIM x
DIM AppToken AS GeodeToken

AppToken.tokenchars = "WP00"
AppToken.manufid = 0
x = GeoRunTestRunning(AppToken)
IF x THEN MsgBox "GeoWrite läuft schon"
```

### GeoRunCloseAppliaction

GeoRunCloseApplication schließt eine laufende GEOS-Applikation. Genauer: sie wird aufgefordert sich zu schließen, genau so, als ob der Nutzer "Beenden" aus dem Datei-Menü gewählt hätte.

Deklaration:

*FUNCTION GeoRunCloseApplication(token as GeodeToken) as Integer*

Parameter:

token: GeodeToken der zu schließenden Applikation.

Return: TRUE: Die Applikation lief und wurde aufgefordert, sich zu beenden.  
FALSE: Die Applikation wurde nicht gefunden (lief nicht).

**Warnung!** GeoRunCloseApplication schließt eine eventuell bestehende Connection zum Programm. Diese Warnung ist nur relevant, wenn Sie GeoRunOpenConnection (siehe unten) verwenden.

Beispiel: GeoDraw schließen

```
DIM AppToken AS GeodeToken

AppToken.tokenchars = "DP00"
AppToken.manufid = 0
GeoRunCloseApplication(AppToken)
```

### 2 Fernsteuern anderer Applikationen

Mit den hier bereitgestellten Routinen können Sie eine Verbindung ("Connection") zu anderen GEOS-Programmen herstellen und diesen Befehle senden. Das können R-BASIC Programme oder andere GEOS-Applikationen (z.B. GeoWrite) sein.

*GeoRunOpenConnection* etabliert eine Connection zu einem anderen Programm. Es liefert einen Word-Wert (ConnectionToken), der von den anderen Routinen benötigt wird.

*GeoRunSendMessage* sendet einen Text und einen Word-Wert an den OnConnection-Handler eines verbundenen R-BASIC Programms.

*GeoRunSendMessageBack* sendet einen Text und einen Word-Wert an den OnConnection-Handler des Programms, dass die Verbindung hergestellt hat.

*GeoRunSendKbdEvent* simuliert einen Tastendruck für das verbundene Programm.

*GeoRunCloseConnection* schließt eine Verbindung zu einem anderen Programm wieder.

Während die Connection aktiv ist, können Sie mit **GeoRunStartApplication** oder **GeoRunOpenFile** (siehe vorn) eine Datendatei an das verbundene Programm senden.

#### GeoRunOpenConnection

GeoRunOpenConnection etabliert eine Connection zu einem anderen Programm. Dieses Programm wird als Server bezeichnet, während das aktuelle Programm (dass GeoRunOpenConnection aufgerufen hat) der Client ist. Wenn der Server noch nicht läuft, wird er gestartet. Während die Connection besteht, lässt sich der Server nicht beenden.

GeoRunOpenConnection liefert einen Word-Wert (ConnectionToken), der von den anderen Routinen benötigt wird.

Deklaration:

```
FUNCTION GeoRunOpenConnection(token as GeodeToken, data$ as  
    String(32)) as Integer
```

Parameter:

token: GeodeToken des Servers.

data\$: Name einer Datendatei aus dem aktuellen Ordner.  
Ein Leerstring (keine Datei) ist zulässig.

Return: ConnectionToken für die Verbindung.



## R-BASIC - Extensions: GeoRun Library

Einfach unter PC/GEOS programmieren

**Warnung!** Wenden Sie weder GeoRunCloseApplication noch GeoRunTest-Running auf den Server an, solange die Connection besteht! Diese Routinen schließen die bestehende Verbindung!

GeoRunStartApplication und GeoRunOpenFile hingegen können Sie gefahrlos verwenden.

Hinweise:

- Falls der Server schon läuft ruft GeoRunOpenConnection den OnConnection-Handler des Servers auf, andernfalls den OnStartup-Handler.
- Beachten Sie, dass das Öffnen einer Applikation und/oder das Laden eines Dokuments einen Moment dauern kann.  
Falls Sie in dieser Situation direkt nach GeoRunOpenConnection einen Befehl senden, kann folgendes passieren:  
GeoRunSendMessage: der OnConnection-Handler wird vor dem OnStartup-Handler (gerufen von GeoRunOpenConnection) ausgeführt.  
GeoRunSendKbdEvent: der Tastendruck könnte ignoriert werden.

Beispiel: siehe GeoRunSendMessage und GeoRunSendKbdEvent

### GeoRunSendMessage

GeoRunSendMessage sendet eine Information an den OnConnection-Handler des Servers. Sie besteht aus einem Text mit bis zu 250 Zeichen, der im Parameter dataFile\$ an den Handler übergeben wird und einem Word-Wert, der im Parameter flags an den Handler übergeben wird. Es ist Aufgabe des Servers, die Informationen korrekt zu interpretieren.

GeoRunSendMessage setzt voraus, dass der Server ein R-BASIC Programm ist, dass mit R-BASIC 1.0 oder höher compiliert wurde.

Deklaration

```
SUB GeoRunSendMessage(cTok as Word, data$ as String(250), flags as Word)
```

Parameter

- cTok: ConnectionToken, geliefert von GeoRunOpenConnection
- data\$: beliebiger Text. Wird als Parameter dataFile\$ an den OnConnection-Handler übergeben.
- flags: beliebiger Word-Wert. Wird als Parameter flags an den OnConnection-Handler übergeben.

Hinweise:

- Falls Sie mit GeoRunSendMessage eine den Namen einer Datendatei an den OnConnection-Handler übergeben wollen, sollten Sie den kompletten Pfad zu Datei übergeben.  
Einfacher ist die Verwendung von GeoRunStartApplication oder GeoRunOpenFile für diesen Zweck.
- Falls Sie sowohl Datendateien als auch andere Kommandos an den Server

## R-BASIC - Extensions: GeoRun Library

Einfach unter PC/GEOS programmieren

senden wollen, können Sie Ihre Kommandos mit einem Zeichen beginnen, das nicht Teil eines Dateinamens sein kann, z.B. mit einem Sternchen \*. Damit kann der OnConnection-Handler beides sicher unterscheiden. Das der Library beiliegende Beispiel macht davon Gebrauch.

Beispiel: Wir stellen eine Verbindung zu einem anderen Programm (dem Server) her und senden ihm das ConnectionToken. Der Server muss die Message in seinem OnConnection-Handler korrekt interpretieren. Der Code stammt aus dem der Library beiliegende Beispiel.

```
DIM AppToken AS GeodeToken
DIM globalCTok

AppToken.tokenchars = "BOOK"      ' Muss zum Server passen
AppToken.manufid = 5

' "*1" sagt dem Server, dass der 3. Parameter das
' ConnectionToken ist
globalCTok = GeoRunOpenConnection(AppToken, "")
GeoRunSendMessage(globalCTok, "*1", globalCTok)
```

### GeoRunSendMessageBack

GeoRunSendMessageBack kann vom Server aus gerufen werden und sendet eine Information an den OnConnection-Handler des Clients. Das benötigte ConnectionToken muss dem Server vorher mit GeoRunSendMessage übermittelt worden sein.

Es ist Aufgabe des Clients, die gesendete Informationen korrekt zu interpretieren.

#### Deklaration

```
SUB GeoRunSendMessageBack(cTok as Word, data$ as String(250), flags as Word)
```

#### Parameter

cTok: ConnectionToken, geliefert von GeoRunOpenConnection

data\$: beliebiger Text. Wird als Parameter dataFile\$ an den OnConnection-Handler übergeben.

flags: beliebiger Word-Wert. Wird als Parameter flags an den OnConnection-Handler übergeben.

## GeoRunSendKbdEvent

GeoRunSendKbdEvent simuliert einen Tastendruck für den Server. Der Tastendruck wird an das Application-Objekt übergeben und geht von dort direkt an das aktuell "aktive" Objekt.

Die Parameter sind die gleichen, wie bei der Methode KbdEventWithScanCode. Dort, in den Kapiteln 14.2 und 14.3 des Handbuchs "Spezielle Themen" (Vol. 3), finden Sie eine ausführliche Beschreibung der Bedeutung der einzelnen Parameter.

GeoRunSendKbdEvent kann mit allen GEOS-Applikationen verwendet werden. Bitte beachten Sie die Hinweise unten.

### Deklaration

```
SUB GeoRunSendKbdEvent(cTok as Word, character as Word, keyState as Word, keyFlags as Word, scanCode as Word)
```

### Parameter:

cTok: ConnectionToken, geliefert von GeoRunOpenConnection  
character: ASCII-Code oder erweiterter ASCII-Code der Taste  
Tipp: In der Library "KeyCodes" finden Sie symbolische Konstanten für die erweiterten ASCII-Codes.  
keyState: Information, welche Status- oder Steuertasten aktuell gedrückt sind, z.B. KS\_LSHIFT für die linke Shift-Taste.  
keyFlags: Information, ob der Nutzer die Taste gerade gedrückt hat, sie gedrückt hält oder gerade losgelassen hat, z.B. KF\_FIRST\_PRESS  
scanCode: Der Scancode der gedrückten Taste. Dieser Wert kann Null sein, es sei denn, Sie wollen ein Tastenkürzel wie "Strg-P" aktivieren.

### Hinweise:

- Im Allgemeinen ist es ausreichend, das "FIRST\_PRESS" Event zu senden.
- Menüpunkte, die über eine "Shortcut" Kombination wie Strg-P oder Shift-F4 angesprochen werden, können Sie mit GeoRunSendKbdEvent aktivieren.
- Menüs oder Menüpunkte, die über die normale Tastaturnavigation, z.B. Alt-d, angesprochen werden, können Sie mit GeoRunSendKbdEvent **nicht** aktivieren.

Beispiel: Wir stellen eine Verbindung zu GeoWrite her und senden das Wort "Hallo" an das aktuelle Dokument. Dann öffnen wir den Drucken-Dialog (Strg-P). Für diesen Zweck benötigen wir den Scancode des Buchstaben P. Am Ende des Kapitels 14.3 im Handbuch "Spezielle Themen" finden wir dafür den Wert 25.

```
DIM cTok
DIM AppToken AS GeodeToken

AppToken.tokenchars = "WP00"
AppToken.manufid = 0

cTok = GeoRunOpenConnection(AppToken, "")

GeoRunSendKbdEvent(cTok, Asc("H"), 0, KF_FIRST_PRESS, 0)
```

## R-BASIC - Extensions: GeoRun Library

Einfach unter PC/GEOS programmieren

```
GeoRunSendKbdEvent(cTok, Asc("a"), 0, KF_FIRST_PRESS, 0)
GeoRunSendKbdEvent(cTok, Asc("l"), 0, KF_FIRST_PRESS, 0)
GeoRunSendKbdEvent(cTok, Asc("l"), 0, KF_FIRST_PRESS, 0)
GeoRunSendKbdEvent(cTok, Asc("o"), 0, KF_FIRST_PRESS, 0)

GeoRunSendKbdEvent(cTok, Asc("p"), KS_LCTRL, KF_FIRST_PRESS,
25)

GeoRunCloseConnection(cTok)
```

### GeoRunCloseConnection

GeoRunCloseConnection schließt eine mit GeoRunOpenConnection hergestellte Verbindung.

Deklaration

*SUB GeoRunCloseConnection(cTok as Word)*

Parameter:

cTok: ConnectionToken, geliefert von GeoRunOpenConnection

The GeoRun Library allows the R-BASIC programmer to communicate with other GEOS applications. This may be both R-Apps (compiled R-BASIC programs) and other GEOS applications, such as GeoWrite. A data file can be passed if required.

To use the GeoRun Library you have to include it:

```
Include "GeoRun"
```

## 1 Start and Stop Other Applications

|                        |  |
|------------------------|--|
| GeoRunStartApplication | Passes a file to an application whose GeodeToken you know. If needed, the application will be started. |
| GeoRunOpenFile         | Opens a data file, determining the associated application from the type of the file.                   |
| GeoRunFindDosApp       | Determines the GEOS application associated with a DOS file.  |
| GeoRunTestRunning      | Checks whether a given GEOS application is already running.  |
| GeoRunCloseAppliation  | Closes a running GEOS application.   |

### GeoRunStartApplication

GeoRunStartApplication opens a GEOS application and passes it a data file. You have to be sure that the type of the file matches the application. You can also use GeoRunStartApplication to pass a (further) file to an already running application.

Declaration:

```
FUNCTION GeoRunStartApplication(appToken as GeodeToken, dataFile as String(32)) as WORD
```

Parameters:

appToken: GeodeToken of the application to be opened.

If the application is not running yet, it will be started. To be able to do this, it must be located in the World directory, one of its subdirectories, or in the SysAppl directory.

dataFile: Name of the file to be passed. The file must be located in the current folder and must match the application to be started.

An empty string is allowed.

Return: Null or error code. The global variable fileError is set. See below.

# R-BASIC - GeoRun Library

The simple PC/GEOS Programming Language

Remarks:

- If the associated program is an R-BASIC program, the file is passed to the OnStartup handler if the program would have to be started. If it is already running, the file is passed to the OnConnection handler.
- GeoRunStartApplication (like any function in R-BASIC) can also be used like a SUB. See example.

## Error code details

GeoRunStartApplication and GeoRunOpenFile return zero if the application could be started successfully or is already running. Otherwise, the routines return an error code. In addition, these routines set the global variable fileError (null or error code).

Positive error codes and error codes in the range of -50 to zero are R-BASIC "standard" codes. You can use the routine ErrorText\$() to get a description of the error. Typical error messages are:

ERROR\_FILE\_NOT\_FOUND: The data file does not exist in the current folder.

ERROR\_ACCESS\_DENIED: Access has been denied. This error also occurs if you specify a folder instead of a file.

Error codes -51 and -52 are library-internal errors. They are defined as follows:

```
CONST LIB_ERROR_OPENING_APP      = -51
  ! The application could not be started, mostly because
  ! no application with the passed token was found.
CONST LIB_ERROR_NO_DOS_APP       = -52
  ! No matching DOS application was found in the GEOS.INI file.
  ! Also a *.* - entry does not exist.
```

Example: Opening GeoWrite with a file in the Document folder

```
DIM AppToken AS GeodeToken

AppToken.tokenchars = "WP00"
AppToken.manufid = 0
SetStandardPath(SP_DOCUMENT)

GeoRunStartApplication(appToken, "Funny Faces")
```

## GeoRunOpenFile

GeoRunOpenFile passes a file to the associated application so that it can open it. The file must be located in the current folder. For VM files (e.g. GeoWrite documents) and GEOS data files the associated application is determined from the file type (Creator token), for DOS files the associated application is read from the GEOS.INI ( [fileManager] : filenameTokens).

Declaration:

```
FUNCTION GeoRunOpenFile(dataFile as String(32)) as WORD
```

Parameters:

dataFile: Name of the file to open. The file must be located in the current folder. The associated application is determined automatically.

Return: Zero or error code. The global variable fileError is set. See also: GeoRunStartApplication.

Remarks:

- If the associated program is an R-BASIC program, the file is passed to the OnStartup handler if the program would have to be started. If it is already running, the file is passed to the OnConnection handler.
- GeoRunOpenFile can (like any function in R-BASIC) also be used like a SUB.
- If you don't want the default assignment "\*".\*" to be used for a DOS file for which no explicit assignment exists in GEOS.INI, use GeoRunFindDosApp to check if an explicit assignment exists in GEOS.INI.
- GeoRunOpenFile can also launch an application if it is located in the current folder.

Example: Opening a graphic file in the document folder

```
SetStandardPath(SP_DOCUMENT)  
GeoRunOpenFile("MANJUL.GIF")
```

## GeoRunFindDosApp

GeoRunFindDosApp scans the GEOS.INI file for the application associated with a DOS file and returns the associated GeodeToken.

Declaration:

```
FUNCTION GeoRunFindDosApp(dataFile as String(32), findStar as Integer) as  
GeodeToken
```

Parameters:

dataFile: Complete name of the DOS file. If you only know the file extension, pass a fake name here, e.g. "X.EXT" or "\*.EXT".

## R-BASIC - GeoRun Library

The simple PC/GEOS Programming Language

Never pass only the file extension! This will be interpreted as a file name without extension!

findStar: TRUE or FALSE

TRUE: If no explicit association was found, the token of the "\*.\*" association is returned - if such a association exists.

FALSE: If no explicit association was found a null token is returned. The association to \*.\* is ignored.

Return: GeodeToken of the associated application, or a null token if no associated application was found.

Example: Finding the default browser (HTML viewer)

```
DIM AppToken AS GeodeToken
  appToken = GeoRunFindDosApp("*.HTM", FALSE)
  MsgBox appToken.tokenChars + ", " + Str$(appToken.manufID)
```

Tip:

- To check if a null token was returned, you can use the NullStruct() function.

```
DIM appToken as GeodeToken
  appToken = GeoRunFindDosApp("MYFILE.DTF", FALSE)
  IF appToken = NullStruct() THEN ....
```

### **GeoRunTestRunning**

GeoRunTestRunning checks whether a given application is already running or not.

Declaration:

*FUNCTION GeoRunTestRunning(token as GeodeToken) as Integer*

Parameters:

token: GeodeToken of the application in question.

Return: TRUE: The application is already running.  
FALSE: The application is not running yet.

Warning. GeoRunTestRunning closes a possibly existing connection to the tested program. This warning is only relevant if you use GeoRunOpenConnection (see below).



Example: Check if GeoWrite is already running

```
DIM x
DIM AppToken AS GeodeToken

AppToken.tokenchars = "WP00"
AppToken.manufid = 0
x = GeoRunTestRunning(AppToken)
IF x THEN MsgBox "GeoWrite is running"
```

## GeoRunCloseAppliacion

GeoRunCloseApplication closes a running GEOS application. More exactly: it will be asked to close, just as if the user had chosen "Exit" from the file menu.

Declaration:

*FUNCTION GeoRunCloseApplication(token as GeodeToken) as Integer*

Parameters:

token: GeodeToken of the application to be closed.

Return: TRUE: The application was running and was asked to close.

FALSE: The application was not found (was not running).

**Warning:** GeoRunCloseApplication closes a possibly existing connection to the program. This warning is only relevant if you use GeoRunOpen-Connection (see below).

Example: Close GeoDraw

```
DIM AppToken AS GeodeToken

AppToken.tokenchars = "DP00"
AppToken.manufid = 0
GeoRunCloseApplication(AppToken)
```

## 2 Remote Control of Other Applications

With the routines provided here you can establish a connection to other GEOS programs and send them commands. These can be R-BASIC programs or other GEOS applications (e.g. GeoWrite).

**GeoRunOpenConnection** establishes a connection to another program. It returns a Word value (ConnectionToken), which is needed by the other routines.

**GeoRunSendMessage** sends a text and a Word value to the OnConnection handler of a connected R-BASIC program.

**GeoRunSendMessageBack** sends a text and a Word value to the OnConnection handler of the program that has established the connection.

**GeoRunSendKbdEvent** simulates a keystroke for the connected program.

**GeoRunCloseConnection** closes a connection to another program.

While the connection is active, you can also send a data file to the connected program using **GeoRunStartApplication** or **GeoRunOpenFile** (see ahead).

### GeoRunOpenConnection

**GeoRunOpenConnection** establishes a connection to another program. This program is called the server, while the current program (that called **GeoRunOpenConnection**) is the client. If the server is not running yet, it will be started. While the connection is established, the server cannot be closed.

**GeoRunOpenConnection** returns a Word value (ConnectionToken) which is needed by the other routines.

Declaration:

```
FUNCTION GeoRunOpenConnection(token as GeodeToken, data$ as  
String(32)) as Integer
```

Parameters:

token: GeodeToken of the server.

data\$: Name of a data file located in the current folder.  
An empty string (no file) is allowed.

Return: ConnectionToken for the connection.

**Warning:** Do not apply **GeoRunCloseApplication** or **GeoRunTest-Running** to the server as long as the connection is established! These routines close the existing connection!

**GeoRunStartApplication** and **GeoRunOpenFile** however you can be used safely.

## Remarks:

- If the server is already running, GeoRunOpenConnection calls the server's OnConnection handler, otherwise it calls the OnStartup handler.
- Note that opening an application and/or loading a document may take a moment.

In this situation, if you send a command immediately after GeoRunOpenConnection, the following may happen:

GeoRunSendMessage: the OnConnection handler is executed before the OnStartup handler (called by GeoRunOpenConnection).

GeoRunSendKbdEvent: the keystroke could be ignored.

Example: see GeoRunSendMessage and GeoRunSendKbdEvent.

## **GeoRunSendMessage**

GeoRunSendMessage sends an information to the OnConnection handler of the server. The message consists of a text of up to 250 characters passed to the handler in the dataFile\$ parameter and a Word value passed to the handler in the flags parameter. It is up to the server to interpret the information correctly.

GeoRunSendMessage requires that the server is an R-BASIC program compiled with R-BASIC 1.0 or later.

### Declaration

```
SUB GeoRunSendMessage(cTok as Word, data$ as String(250), flags as Word)
```

### Parameters

cTok: ConnectionToken, returned by GeoRunOpenConnection.

data\$: Any text. Passed as parameter dataFile\$ to the OnConnection handler.

flags: Any word value. Passed as flags parameter to the OnConnection handler.

## Remarks:

- If you want to use GeoRunSendMessage to pass a the name of a data file to the OnConnection handler, you should pass the full path to file. It is easier to use GeoRunStartApplication or GeoRunOpenFile for this purpose.
- If you want to send both data files and other commands to the server, you can start your commands with a character that cannot be part of a file name, e.g. with an asterisk \*. This way the OnConnection handler can safely distinguish between both. The example included with the library uses this.

Example: We connect to another program (the server) and send it the ConnectionToken. The server must interpret the message correctly in its OnConnection handler. The code is taken from the example included in the library.

```
DIM AppToken AS GeodeToken  
DIM globalCTok
```

# R-BASIC - GeoRun Library

The simple PC/GEOS Programming Language

```
AppToken.tokenchars = "BOOK"      ' Must be the servers token
AppToken.manufid = 5

' "*1" tells the Server, that the 3rd parameter is the
' ConnectionToken
globalCTok = GeoRunOpenConnection(AppToken, "")
GeoRunSendMessage(globalCTok, "*1", globalCTok)
```

## **GeoRunSendMessageBack**

GeoRunSendMessageBack can be called from the server and sends an information back to the client's OnConnection handler. The required ConnectionToken must have been previously sent to the server with GeoRunSendMessage.

It is up to the client to interpret the transmitted information correctly.

### Declaration

```
SUB GeoRunSendMessageBack(cTok as Word, data$ as String(250), flags as
Word)
```

### Parameters

cTok: ConnectionToken, returned by GeoRunOpenConnection.

data\$: Any text. Passed as parameter dataFile\$ to the OnConnection handler.

flags: Any word value. Passed as flags parameter to the OnConnection handler.

## **GeoRunSendKbdEvent**

GeoRunSendKbdEvent simulates a keystroke for the server. The keystroke is passed to the Application object and from there it goes directly to the currently "active" object.

The parameters are the same as for the method KbdEventWithScancode. There, in chapters 14.2 and 14.3 of the manual "Special Topics" (Vol. 3), you will find a detailed description of the meaning of the individual parameters.

GeoRunSendKbdEvent can be used with all GEOS applications. Please note the remarks below.

### Declaration

```
SUB GeoRunSendKbdEvent(cTok as Word, character as Word, kexState as
Word, keyFlags as Word, scanCode as Word)
```

### Parameters:

cTok: ConnectionToken, returned by GeoRunOpenConnection

character: ASCII code or extended ASCII code of the key  
Tip: In the library "KeyCodes" you can find symbolic constants for the extended ASCII codes.

keyState: Information about which status or control keys are currently pressed, e.g. KS\_LSHIFT for the left shift key.

keyFlags: Information whether the user has just pressed the key, is holding it down, or has just released it, e.g. KF\_FIRST\_PRESS.

scanCode: The scan code of the pressed key. This value can be zero, unless you want to activate a keyboard shortcut like "Ctrl-P".

## Remarks:

- In most cases it is sufficient to send the "FIRST\_PRESS" event.
- Menu items, which are addressed by a "shortcut" combination like Ctrl-P or Shift-F4, can be activated with GeoRunSendKbdEvent.
- You **cannot** activate menus or menu items that are accessed via normal keyboard navigation, such as Alt-d, with GeoRunSendKbdEvent.

Example: We connect to GeoWrite and send the word "Hello" to the current document. Then we open the print dialog (Ctrl-P). For this purpose we need the scancode of the letter P. At the end of chapter 14.3 in the manual "Special Topics" we find the value 25 for this.

```
DIM cTok
DIM AppToken AS GeodeToken

AppToken.tokenchars = "WP00"
AppToken.manufid = 0

cTok = GeoRunOpenConnection(AppToken, "")

GeoRunSendKbdEvent(cTok, Asc("H"), 0, KF_FIRST_PRESS, 0)
GeoRunSendKbdEvent(cTok, Asc("a"), 0, KF_FIRST_PRESS, 0)
GeoRunSendKbdEvent(cTok, Asc("l"), 0, KF_FIRST_PRESS, 0)
GeoRunSendKbdEvent(cTok, Asc("l"), 0, KF_FIRST_PRESS, 0)
GeoRunSendKbdEvent(cTok, Asc("o"), 0, KF_FIRST_PRESS, 0)

GeoRunSendKbdEvent(cTok, Asc("p"), KS_LCTRL, KF_FIRST_PRESS,
25)

GeoRunCloseConnection(cTok)
```

## GeoRunCloseConnection

GeoRunCloseConnection closes a connection, established with GeoRunOpenConnection.

## R-BASIC - GeoRun Library

The simple PC/GEOS Programming Language

---

Declaration

*SUB GeoRunCloseConnection(cTok as Word)*

Parameters:

cTok: ConnectionToken, returned by GeoRunOpenConnection