

GPR400
Interface Library
Programmer's Guide

Version 2.0

GEMPLUS

July, 1996

At the time of going to press, this guide is as thorough and correct as possible; however, information herein contained may have been updated after this date. GEMPLUS reserves the right to change the functions and specifications of its products at any time without prior notice.

This document was prepared by GEMPLUS for both its clients and for its own internal use. Information herein contained is the sole property of GEMPLUS and shall not under any circumstances be reproduced without prior consent of the company.

© Copyright GEMPLUS, 1996.

Smart Cards and Smart Card readers are patent protected by INNOVATRON and Bull CP8 and are produced by GEMPLUS under license.

MS-DOS®, Visual Basic® and Windows® are registered trademarks of Microsoft Corporation
Borland® is a registered trademark of Borland Corporation

Printed in France.

GEMPLUS, B.P. 100, 13881 GEMENOS CEDEX, FRANCE.

Tel: +33 (0)4.42.36.50.00 Fax: +33 (0)4.42.36.50.90

Document Reference : E5234016/DPD03810B00

ABOUT THIS GUIDE

This guide provides information on how to install and use the GPR400 Interface Library to develop smart cards' applications for the GEMPLUS GPR400 Smart Card Reader.

Audience

This guide assumes that you are familiar with smart card concepts and terminology, and the ISO 7816-3 and -4 standards. It also assumes that you have programming experience in C language and Visual Basic.

How to Use This Guide

The following paragraphs tell you where to find information when you need it. It is important that you read this section in order to use this Guide to its full potential.

Developing GPR400 Library Applications

Read this section for a general description of the GPR400 Interface Library, and its development environments.

GPR400 Interface Library Parameters

Read this section for the communication parameters that apply to the GPR400 and their identifiers.

GPR400 Commands

Read this section for a description of the GPR400 Interface Library commands.

Visual Basic Utilities

Read this section for a description of the utilities required for using the library with Visual Basic.

Status Codes

Read this section for a description of status codes that may be returned by the GCR Interface Library.

APDU Command Format

Read this section for a description of the format of the commands used for reader and card exchange.

TABLE OF CONTENTS

ABOUT THIS GUIDE	i
Audience	i
How to Use This Guide.....	i
TABLE OF CONTENTS	ii
GLOSSARY	iv
OVERVIEW	1
DEVELOPING GPR400 LIBRARY APPLICATIONS	2
Layer Structure	2
Application Development Sequence	3
Status Codes.....	3
Parameters Structures.....	3
Include Files and Libraries.....	4
GPR400 INTERFACE LIBRARY PARAMETERS	7
Reader Parameters.....	7
Types of Readers	8
Types of Protocol.....	8
Session Parameters	9
Identifiers.....	9
Types of Cards	10
Card Exchanges	12
Reader Status	14
Fields Size.....	15
GPR400 COMMANDS	16
GPR400 Generic Commands.....	18
GPR400 Specific Commands	36
VISUAL BASIC UTILITIES	46
Visual Basic Utilities for Windows	46
APPENDIX A. STATUS CODES	50
Host Status Codes	50
Host/Reader Status Codes	51
Reader Status Codes	52
Reader/Card Status Codes	53

Card Status Codes.....	54
APDU Status Codes.....	54
TLV Status Codes.....	55
File Management Status Codes.....	55
Warnings.....	55
Other Errors.....	56
APPENDIX B. APDU COMMAND FORMAT	57
Command APDU.....	57
Response APDU.....	58
APDU Types.....	58
APPENDIX C.	59

GLOSSARY

A.P.D.U.:	Application Protocol Data Unit
A.P.I.:	Application Programming Interface
Card Services:	Soft layer above the Socket services coordinating the allocation of system resources amongst the cards in the socket
C.I.S.:	Card Information Structure; data structure containing information about the format and the organisation of the data on the PCCARD
I.C.C.:	Integrated Circuit Card
I.F.D.:	Interface Device
PCCARD:	Memory of IO card compatible with the card standard
P.C.M.C.I.A.:	Personal Computer Memory Card International Association
Socket Services:	Software that talks directly to the Hardware and provides a standardized interface for PCCARDS, sockets and adaptors to hide hardware specifics from drivers.
T.P.D.U.:	Transport Protocol Data Unit
T.L.V.:	Command with a Tag Length Value format

OVERVIEW

The GPR400 Interface Library is a library of commands that interface with smart cards through the GEMPLUS GPR400 Smart Card Reader. You call GPR400 Interface Library commands from applications written in the C or Visual Basic 3 and 4 programming languages, running under DOS, Windows 3.xx and Windows 32 bits environments.

The GPR400 Interface Library has been designed to be as independent as possible from the cards' operating system. It can communicate with microprocessor cards and also with any kind of synchronous cards or custom cards.

This guide describes how to develop applications using the library commands.

See the GPR400 Installation Guide for PCMCIA Software Environment. You will find drivers and libraries suited to your needs (e.g. specific PCMCIA platforms) on the GEMPLUS server at the following address:

<http://www.gemplus.com/gpr400.html>

See the GPR400 Reference Manual for a list of low level commands built in the GPR400 operating system.

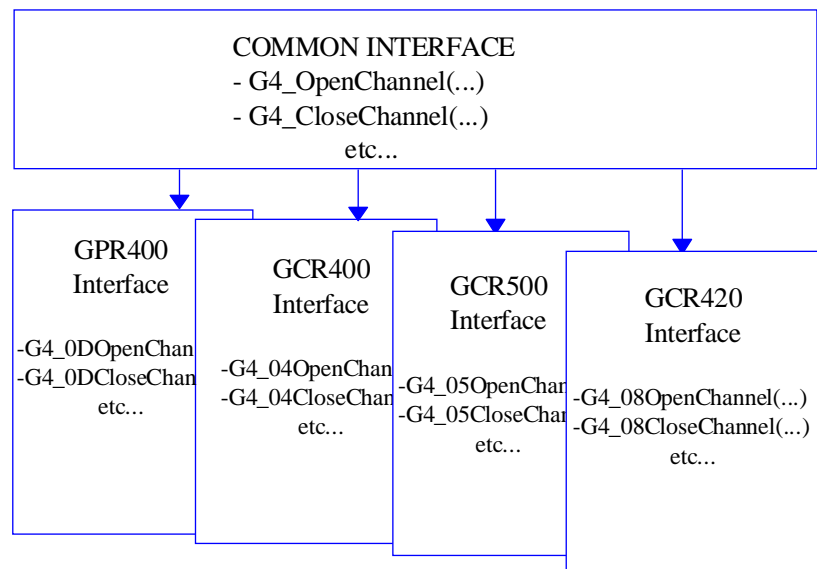
DEVELOPING GPR400 LIBRARY APPLICATIONS

This section outlines the steps that you need to follow in order to develop smart card applications using the GPR400 Interface Library. It describes:

- the general guidelines for developing the GPR400 Interface Library applications
- troubleshooting
- parameter structures
- files and libraries to be used in the development environment

Layer Structure

The API 4 architecture defined for the Gemplus readers, enables the use of the same interface whatever the reader. However, this common interface only specifies functions common to the readers. To make use of functions specific to the reader, the reader interface must be used.



Application Development Sequence

When you develop applications using the GPR400 Interface Library, the application must initially perform the following sequence:

Configure the port that the reader is connected to and define the reader type. Use the *G4_OpenChannel* command.

Now you can send and receive data from the reader.

After this it can:

1. Define the type of card that the application is developed for and reset the card. Use the *G4_OpenSession* command.
2. Send and receive data from the card using the *G4_ExchangeAPDU* command.
3. Close communication sessions using the *G4_CloseSession* command, which powers down the card.
4. Terminate the application using the *G4_CloseChannel* command to close the channel that is open. You cannot use the channel for another application while it remains open.

Status Codes

When a command is executed, the library returns a status command that indicates the result of the command. Error codes are sent when:

- there is a problem with the host PC. For example, the specified port is open or a system error has occurred. If this is the case, a Host status code is returned.
- there is a problem between the host PC and the reader. For example, there was a communication error between the host PC and the reader, or a message is too long. If this is the case, a Host/Reader status code is returned.
- there is a problem with the reader. For example, it is badly connected, or a command cannot be carried out. If this is the case, a Reader status code is returned.
- there is a problem between the reader and the card. For example, the answer to reset is incoherent or an incorrect APDU command was sent to the card. If this is the case, a Reader/Card status code is returned.
- there is a problem with the card. For example, there is no card in the reader, or a card was removed from the reader. If this is the case, a Card status code is returned.

When a command is carried out successfully the library returns a positive or null status code.

All the possible return status codes are listed with a description of their meanings in Appendix A.

Parameters Structures

Many of the GPR400 Interface Library commands refer to and update the structure parameters, which are defined in the GEMGCR.H, GEMGCR.VB3 and GEMGCR.VB4 include files. The parameter structures are used by the GPR400 Interface Library.

Include Files and Libraries

The following paragraphs describe the library and include files that you need for all the GPR400 Interface Library compatible development environments.

Include Files Description

Find below a listing of all include files along with their description:

- GEMPLUS.H: includes the GEMDEF.H, GEMMAC.H and GEMERROR.H files.
- GEMDEF.H, GEMDEF.VB3, GEMDEF.VB4: find there the definition of all the types used in the Gemplus software (e.g. BYTE, WORD16, G_FAR etc.)
- GEMMAC.H: find there the C macros definition (e.g. LOWORD, ComPort etc)
- GEMERROR.H, GEMERROR.VB3, GEMERROR.VB4: store all GEMPLUS error codes definitions (e.g. GE_ICC_ABSENT)
- GEMGCR.H, GEMGCR.VB3, GEMGCR.VB4: include the GEMCOM.H file and hold the structures definition (e.g. G4_APDU_COMM, G4_CHANNEL_PARAM etc.) as well as predefined values and the definition of the API version 4 prototype functions for the Gemplus readers when called through the common core module (e.g. INT16G_DECL G4_OpenChannel etc.).
- GEMCOM.H, GEMCOM.VB3, GEMCOM.VB4: find there the definition of the structures used for the definition of the readers communication layers (e.g. COM_TYPE, COM_SERIAL etc.)
- GEMRDR.H, GEMRDR.VB3, GEMRDR.VB4: store the definition of the values predefined for all the Gemplus readers (e.g. GCR400, GCR500 etc.)
- GEMCARD.H, GEMCARD.VB3, GEMCARD.VB4: store the definition of the values predefined for some of the cards used with Gemplus readers (e.g. ISOCARD, GPM256, etc.)
- PILGR0D.H, PILGR0D.VB3, PILGR0D.VB4: store the definition of the API version 4 prototype functions for the GPR400 reader (e.g. INT16 G_DECL G4_0DOpenChannel)

C Compilers for DOS

Include Files:

GEMPLUS.H, GEMGCR.H

Optional Include Files:

GEMRDR.H, GEMCARD.H, PILGR0D.H

Library Files:

**GCR40xyz.LIB, GR0Dxyz.LIB, GCHANxyz.LIB, EMP04xyz.LIB,
EMP05xyz.LIB, EMP08xyz.LIB, GCSxyz.LIB**

where:

xy: - V1 for Microsoft Visual C++1.xx
- M7 for Microsoft C7
- B5 for Borland C++5.xx
- B4 for Borland C++4.xx

z: - L for large memory model

Example:

GCR40V1L.LIB for Microsoft Visual C++1.xx for large memory model.

Note: EMP....LIB are empty libraries used by the API Version 4 common core.

Compilation - Options-

- **provide a big enough stack in order to prevent stack overflow**
- **add the LINK /NOE option**

C Compilers for Windows 3.1x

Include Files:

GEMPLUS.H, GEMGCR.H

Optional Include Files:

GEMRDR.H, GEMCARD.H, PILGR0D.H

Library Files:

WGCR40.LIB, WGR0D.LIB

DLLs Files:

WGCR40.DLL, WGTCHAN.DLL, WGR0D.DLL

Compilation - Options-

- **2 bytes alignment**

Borland Compilers: With Borland compilers your application must be compiled with Compact or larger memory models. Use the IMPLIB utility to generate the DLLs LIB files.

C Compilers for Windows 32 bits

Include Files:

GEMPLUS.H, GEMGCR.H

Optional Include Files:

GEMRDR.H, GEMCARD.H, PILGR0D.H

Library Files:

W32GCR40.LIB, W32GR0D.LIB

DLLs Files:

W32GCR40.DLL

Compilation - Options-

- 4 bytes alignment

Note: The DLLs specific to the GPR400 are automatically installed during the software installation process.

Visual Basic 3 for Windows 3.1

Include Files:

**GEMDEF.VB3, GEMERROR.VB3, GEMGCR.VB3,
GEMCOM.VB3**

Optional Include Files:

GEMRDR.VB3, GEMCARD.VB3, PILGR0D.VB3

DLLs Files:

WGCR40.DLL, WGTCHAN.DLL, WGR0D.DLL

Visual Basic 4 for Windows 32 bits

Include Files:

**GEMDEF.VB4, GEMERROR.VB4, GEMGCR.VB4,
GEMCOM.VB4**

Optional Include Files:

GEMRDR.VB4, GEMCARD.VB4, PILGR0D.VB4

DLLs Files:

W32GCR40.DLL

Note: The DLLs specific to the GPR400 are automatically installed during the software installation process.

GPR400 INTERFACE LIBRARY PARAMETERS

This section contains definitions and structures for objects managed by the GPR400 Interface Library.

Reader Parameters

Find below a list of parameters definitions for reader communication:

Identifiers

G4_CHANNEL_PARAM

DOS and Windows 3.1

WORD16 IFDType
WORD32 IFDBaudRate
COM_TYPE IFDMode
COM_PCCARD PcCard

Windows 32 bits

WORD32 IFDType
WORD32 IFDBaudRate
COM_TYPE IFDMode
COM_PCCARD PcCard

Where:

IFDType	Holds the connected reader. (See section <i>Type of Readers</i>)
IFDBaudRate	Not used for the GPR400 reader
IFDMode	Selects the reader connection mode; sets the G_PCCARD value (See section <i>Types of Protocols</i>)
PcCard	Holds the communication parameters (See section <i>Types of Protocol</i>)

Types of Readers

The Reader section of the library defines the smart card and security module readers that you can currently use with the interface library.

An identifier can be used as a reference to the GPR400. This does not prevent nominal use of the GPR400 as a variable. The present section defines the identifier for the GPR400 reader.

The identifier is entered at the IFDType field in the G4_CHANNEL_PARAM structure of the G4_OpenChannel function.

Identifier	Value
GPR400	0x0D

These are defined in the GEMRDR include files.

Types of Protocol

The Protocol section of the library defines the protocols that you can currently use between the host and the reader PCCARD.

The union Comm is seen as a COM_PCCARD structure which holds two fields, Socket and ITNumber:

Where:

Socket	is the PCMCIA socket identifier, values from 0 to 16
ITNumber	is used to select the interrupt value that is used to drive the selected port. It is not possible to define the IT number directly with Windows. DEFAULT_IT is used to select the automatic interrupt. Its value is then defined as 0xFF.

This is defined in the GEMCOM include files.

Note: The GPR400 Interface Library does not use an interrupt resource from your host. However, some host platforms need to declare it for the embedded PCCARD architecture. (For more details see the GPR400 Installation Guide).

Session Parameters

Listed below are parameter definitions for a session.

Identifiers

G4_SESSION_PARAM

For DOS and Windows 3.1

```
WORD16 ICCType
WORD32 ApduLenMax
WORD16 ResetLen
WORD16 HistLen
WORD16 HistOffset
WORD8  ResetChar[MAX_RESET_LEN]
WORD8  Dummy[MAX_DUMMY_SESSION]
```

For Windows 32 bits

```
WORD32 ICCType
WORD32 ApduLenMax
WORD32 ResetLen
WORD32 HistLen
WORD32 HistOffset
WORD8  ResetChar[MAX_RESET_LEN]
WORD8  Dummy[MAX_DUMMY_SESSION]
```

where:

ICCType	holds the awaited card type: See section <i>Types of Cards</i>
ApduLenMax	holds the maximal APDU length supported according to the card.
ResetLen	holds the Answer To Reset length.
HistLen	holds the number of available historical characters.
HistOffset	holds offset for the first historical byte in reset character buffer.
ResetChar	holds the Answer To Reset characters.

A dummy field is added for a future use.

Types of Cards

The Card section of the include files define the cards that you can currently use with the interface library.

These are listed below.

Identifiers

The card identifier is entered at the ICCType field in the G4_SESSION_PARAM structure. You can also use the genuine OROS card types.

Card Type	API Code
ISOCARD	0x02
COSCARD	0x02
GFM	0x06
GPM103	0x07
GPM256	0x03
GPM271	0x0E
GPM276	0x0D
GPM416	0x04
GPM416R	0x14
GPM896	0x04
GPM896R	0x14
GPM2K	0x09
GPM8K	0x08
GAM	0x0A
GAM144	0x0A
GAM226	0x0F
GSM1K	0xF4
GSM4K	0xF6

Note 1: GPM416, GPM896 simulate the fuse blow.

GPM416R, GPM896R have a real comportment.

Note 2: ISOCARD and COSCARD are 7816-3 compatible microprocessor cards (T=0, T=1)

Note 3: For synchronous cards (GPM256, GFM, etc...) external drivers are mandatory.

These are defined in the GEMCARD include files.

The cards drivers' names have the following format:

ICCDRVxx.GPR

Where:

xx represents the card type in hexadecimal notation.

Example:

ICCDRV03.GPR is the GPM256 card driver.

For Windows 3.1:

The directory pathname where the card drivers are held, must be declared in the GEMPLUS.INI file which must be located in the Windows directory.

In the GEMPLUS.INI file, there must be the following section:

[GPR]

with the following:

DRVPATH=directory Path

where the carddrivers are located.

For DOS and Windows 32 bits:

The card drivers must be located in the current directory.

Notes: Do not hold several copies of the card drivers in multiple directories in order to avoid eventual updating problems.

During installation of WinGPR software, all the above is executed.

Card Exchanges

Listed below are the parameters for card exchanges for APDU communication and APDU response.

Identifiers

G4_APDU_COMM

For DOS, Windows 3.1, Windows 32 bits

WORD8	Command[COMMAND_LEN]
WORD32	LengthIn
WORD8	G_FAR *DataIn
WORD32	LengthExpected

where:

Command	holds the command bytes to send to card.
LengthIn	holds the number of bytes to send to card. The allowed range is {0 .. 65535}. The NULL_LEN is used for GEMPLUS specific products.
DataIn	holds the bytes to send to card. Warning: the user must allocate this buffer.
LengthExpected	memorizes the maximum length expected in the card response. The allowed range is {0 .. 65536}.

G4_APDU_RESP

For DOS and Windows 3.1

WORD32	LengthOut
WORD8	G_FAR *DataOut
WORD16	Status

For Windows 32 bits

WORD32	LengthOut
WORD8	G_FAR *DataOut
WORD32	Status

where:

LengthOut is the real number of received bytes.

DataOut holds the received bytes. The size of the buffer must be at least the number of bytes expected by the command sent.

Warning: the user must allocate this buffer.

Status holds the two status bytes SW1 and SW2.

Reader Status

Identifiers

G4_IFD_STATUS

For DOS and Windows 3.1

G4_CHANNEL PARAM	Parameters
WORD16	Protocol
WORD32	ExchangeSize
WORD16	OSLength
WORD8	OSString [MAX_IFD_STRING]
WORD16	SpecificLength
WORD8	Specific [MAX_IFD_SPECIFIC]

For Windows 32 bits

G4_CHANNEL PARAM	Parameters
WORD32	Protocol
WORD32	ExchangeSize
WORD32	OSLength
WORD8	OSString [MAX_IFD_STRING]
WORD32	SpecificLength
WORD8	Specific [MAX_IFD_SPECIFIC]

where:

Parameters	stores the communication parameters given to G4_OpenChannel.
Protocol	returns the protocol currently in use with the reader. The GPR400 returns PCCARD_PROTOCOL.
ExchangeSize in the	holds the communication buffer size currently available reader.
OSLength	stores the number of bytes available in the OSString.
OSString	holds a null terminated string that identifies the OS name and version.
SpecificLength	stores the number of updated bytes in the dummy field.

Specific holds various information for the selected

GPR400:

- 128 Kb Flash Memory present or not
- ICC Driver loaded or not
- IC Card inserted or not
- ICC Clock Speed 3.68 or 7.36 Mhz

Fields Size

Listed below are the identifiers for the maximum field lengths and the identifiers for card commands.

Identifiers

```

MAX_RESET_LEN          33
MAX_APDU_LEN           65535
MAX_IFD_STRING         100
MAX_IFD_SPECIFIC      100
MAX_DUMMY_CHANNEL     100
MAX_DUMMY_SESSION     100 - MAX_RESET_LEN

```

where:

```

MAX_RESET_LEN          = maximum length for answer to reset
MAX_APDU_LEN           = maximum length for an APDU buffer. ISO
                        defines this as 65535, but low level
                        drivers are more limited than this.

MAX_IFD_STRING         = maximum length for an IFD identifier.
MAX_IFD_SPECIFIC      = maximum length for the specific IFD
                        field
MAX_DUMMY_CHANNEL     = maximum length in fields reserved for
MAX_DUMMY_SESSION     = future use.

```

For ICC Commands

```

COMMAND_LEN           4
NULL_LEN              0xFFFFFFFF

```

where:

```

COMMAND_LEN           = the size of an ICC command
NULL_LEN              = value used with GEMPLUS ICC products.

```

GPR400 COMMANDS

You will find in this section a listing of all GPR400 commands. The commands belong to three different groups:

- Generic commands: generic commands are common commands that interface with smart cards through all the GEMPLUS card readers.
- Specific commands: specific commands usable only with the GEMPLUS GPR400 Smart Card Reader.
- Visual Basic Utilities: can be found in the VBGTOOL4.DLL for Windows 3.1 and VBTOOL32.DLL for Windows NT and Windows 95.

Note: All GPR400 commands will be listed per type. A comprehensive description of each command will follow.

Command Name	Description	Page
<i>Generic Commands</i>		
G4_OpenChannel	Opens a channel to the reader	18
G4_LockChannel	Locks the access of a port associated to the logical channel number	19
G4_UnlockChannel	Unlocks the access of a port associated to the logical channel number	20
G4_CloseChannel	Closes a previously opened channel	21
G4_OpenSession	Opens a session with a selected card	22
G4_CloseSession	Closes a previously opened session	23
G4_SwitchSession	Opens a warm session with a selected card	24
G4_ExchangeApdu	Sends/receives an APDU command to/from an ICC	25
G4_CmdGetTimeout	Returns selected reader command timeout	26
G4_CmdSetTimeout	Changes the current timeout command value	27
G4_ICCDetection	Installs software check for card detection	28
G4_ICCStatus	Returns card information	29
G4_IFDDownload	Enables data download into IFD	30
G4_IFDStatus	Returns information about the reader	31
G4_IFDExchange	Enables exchanges with the GPR400	32
G4_IFDGetPowerTimeout	Returns power timeout for selected reader	33
G4_IFDSetPowerTimeout	Changes power timeout for selected reader	34
G4_ICCSetProtocol	Changes the protocol in use between IFD/ICC	35
<i>Specific Commands</i>		
G4_ODPowerDown	Puts the reader in power down mode	36
G4_ODStandby	Puts the reader in standby mode	37
G4_ODResetGPR	Wakes up or resets the reader	38
G4_ODSwitchClock	Switches the GPR400 clock frequency	39
G4_ODLoadMemory	Loads data in GPR400 memory	40
G4_ODReadMemory	Reads data from the GPR400 memory	41
G4_0DExecMemory	Calls a loaded code	44
G4_0DActivateDriver	Activates a loaded IC Card driver	42
G4_0DDeactivateDriver	Deactivates a loaded IC Card driver	43
G4_0DGetCis	Returns a string of Card Services and PC Card information	45
<i>Visual Basic Utilities</i>		
G4_StrToPtr	Converts a Visual Basic string to its pointer	46
G4_ByteArrayToPtr	Converts a Visual Basic array of bytes to a long variable that is the array of bytes address	47
G4_WritePCCARDComm	Updates Comm union fields	48
G4_ReadPCCARDComm	Reads Comm union fields	49

GPR400 Generic Commands

G4_OpenChannel

This command opens a reader channel and allocates a logical number to it. You may open as many channels as you like for an application, but you must close each opened channel using the G4_CloseChannel command before the end of the application.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```

C          INT16 G_DECL G4_OpenChannel
          (
          const G4_CHANNEL_PARAM G_FAR *Channel
          );

Visual Basic 3      Declare Function G4_OpenChannel Lib "WGCR40.DLL" (Channel As
for Windows        G4_CHANNEL_PARAM) As Integer
3.1

Visual Basic 4      Declare Function G4_OpenChannel Lib "W32GCR40.DLL" (Channel As
for Windows 32    G4_CHANNEL_PARAM) As Integer
bits

```

Parameters In

Channel.IFDType	identifies the target reader.
Channel.IFDMode	selects the connection mode for the reader.
Channel.Comm	holds the communication parameters.

Response

A positive or null value as channel identifier or negative value as status code identifying an error.

Remarks

The IFDType must have the value 0x0D.

The IFDMode must select the PCMCIA connection mode: G_PCCARD.

The Comm structure holds two communication parameters:

- Socket (value 0 to 16)

ITNumber (0xFF)

G4_LockChannel

This command locks the access of a port associated to a logical channel number. Any other application will be prevented from communicating with the same reader.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

C `INT16 G4_LockChannel`
 (
 const WORD16 ChannelNb,
);

*Visual Basic
 3 for
 Windows 3.1* `Declare Function G4_LockChannel Lib "WGCR40.DLL" (ByVal ChannelNb As Integer) As Integer`

*Visual Basic
 4 for
 Windows 32
 bits* `Declare Function G4_LockChannel Lib "W32GCR40.DLL" (ByVal ChannelNb As Integer) As Integer`

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
-----------	---

Response

See Appendix A for a full list of status codes.

G4_UnLockChannel

This command unlocks the access of a port associated to a logical channel number.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C          INT16 G4_UnLockChannel
           (
           const WORD16 ChannelNb,
           );

Visual Basic
3 for
Windows 3.1
           Declare Function G4_UnLockChannel Lib "WGCR40.DLL" (ByVal ChannelNb
           As Integer) As Integer

Visual Basic
4 for
Windows 32
bits
           Declare Function G4_UnLockChannel Lib "W32GCR40.DLL" (ByVal
           ChannelNb As Integer) As Integer
```

Parameters In

ChannelNb indicates the logical number associated during open function to the selected channel.

Response

See Appendix A for a full list of status codes.

G4_CloseChannel

This command closes a previously opened reader channel. This command must be executed in order to free the open port for another application.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C          INT16 G_DECL G4_CloseChannel
          (
          const WORD16 ChannelNb
          );
```

***Visual Basic 3
for Windows
3.1***

```
Declare Function G4_CloseChannel Lib "WGCR40.DLL" (ByVal ChannelNb
As Integer) As Integer
```

***Visual Basic
4 for
Windows 32
bits***

```
Declare Function G4_CloseChannel Lib "W32GCR40.DLL" (ByVal ChannelNb
As Integer) As Integer
```

Parameters In

ChannelNb Logical number allocated to the channel to close

Response

See Appendix A for a full list of status codes.

G4_OpenSession

This command opens a session with a selected card type, and updates the session structure with the values returned by the card Answer to reset. The card is previously powered down.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format*C*

```
INT16 G4_OpenSession
(
  const WORD16 ChannelNb,
  G4_SESSION_PARAM G_FAR *Session
);
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_OpenSession Lib "WGCR40.DLL" (ByVal Nb As Integer, Session As G4_SESSION_PARAM) As Integer
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_OpenSession Lib "W32GCR40.DLL" (ByVal Nb As Integer, Session As G4_SESSION_PARAM) As Integer
```

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
Session.ICCType	is set with the card which will be used during the session.

Parameters Out

Session.ApduLenMax	holds the maximal APDU length supported according to card.
Session.ResetLen	holds the Answer To Reset length.
Session.HistLen	holds the number of available historical characters.
Session.HistOffset	holds offset for the first historical byte in the Reset Character buffer.
Session.ResetChar	holds the Answer To Reset characters.

Response

See Appendix A for a full list of status codes.

Remarks

For some cards, this command downloads a specific driver from a directory specified in the GEMPLUS.INI file. For more information see section *Card Types*.

G4_CloseSession

This command closes a previously opened session and powers off the card.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C          INT16 G4_CloseSession  
          (  
          const WORD16 ChannelNb,  
          );
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_CloseSession Lib "WGCR40.DLL" (ByVal ChannelNb  
As Integer) As Integer
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_CloseSession Lib "W32GCR40.DLL" (ByVal ChannelNb  
As Integer) As Integer
```

Parameters In

ChannelNb indicates the logical number allocated to the channel whose session you want to close.

Response

See Appendix A for a full list of status codes.

G4_SwitchSession

This command opens a warm session with a selected card type, and updates the session structure with the values returned by the card Answer to Reset. A reset is sent to the card without any attempt to power it down. This command is not supported by all cards.

This command is only available when a session is already open.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format*C*

```
INT16 G4_SwitchSession
(
  const WORD16 ChannelNb,
  G4_SESSION_PARAM G_FAR *Session
);
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_SwitchSession Lib "WGCR40.DLL" (ByVal Nb As Integer, Session As G4_SESSION_PARAM) As Integer
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_SwitchSession Lib "W32GCR40.DLL" (ByVal Nb As Integer, Session As G4_SESSION_PARAM) As Integer
```

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
Session.ICCType	is set with the card which will be used during the session.

Parameters Out

Session.ApduLenMax	holds the maximal APDU length supported according to card.
Session.ResetLen	holds the Answer To Reset length.
Session.HistLen	holds the number of available historical characters.
Session.HistOffset	holds offset for the first historical byte in Reset Character buffer.
Session.ResetChar	holds the Answer To Reset characters.

Response

See Appendix A for a full list of status codes.

Remarks

The cards used during the session are of the same type as specified in the G4_OpenSession.

G4_ExchangeApu

This command sends an APDU command to a card via the open channel and returns the card's response.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C
INT16 G_DECL G4_ExchangeApu
(
  const WORD16 ChannelNb,
  G4_APDU_COMM G_FAR *ApuComm,
  G4_APDU_RESP G_FAR *ApuResp
)
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_ExchangeApu Lib "WGCR40.DLL" (ByVal ChannelNb
As Integer, ApuComm As G4_APDU_COMM, ApuResp As G4_APDU_RESP) As
Integer
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_ExchangeApu Lib "W32GCR40.DLL" (ByVal ChannelNb
As Integer, ApuComm As G4_APDU_COMM, ApuResp As G4_APDU_RESP) As
Integer
```

Parameters In

ChannelNb	indicates the logical number allocated to the channel.
ApuComm.Command[1]	CLA - Instruction class
ApuComm.Command[2]	INS- Instruction code
ApuComm.Command[3]	P1- Parameter 1
ApuComm.Command[4]	P2- Parameter 2
ApuComm.LengthIn	Number of characters sent to the card (APDU Lc field)
ApuComm.DataIn	Data buffer to send to the card
ApuComm.LengthExpected	Maximum expected response length (APDU Le field)

Parameters Out

This command updates the following parameters:

ApuResp.LengthOut	Actual length of the card response
ApuResp.DataOut	Buffer of sufficient size to hold the response APDU
ApuResp.Status	The status bytes SW1, SW2 returned by the card

Response

See Appendix A for a full list of status codes.

G4_CmdGetTimeout

This command reads the timeout value currently in use. This is the time needed by the reader to respond to the command. It varies according to the card in use. The default value is 3000 ms.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

C

```
INT16 G_DECL G4_CmdGetTimeout
(
  const WORD16 ChannelNb,
  WORD32 G_FAR *Timeout
)
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_CmdGetTimeout Lib "WGCR40.DLL" (ByVal ChannelNb
As Integer, Timeout As Long) As Integer
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_CmdGetTimeout Lib "W32GCR40.DLL" (ByVal
ChannelNb As Integer, Timeout As Long) As Integer
```

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
-----------	---

Parameters Out

Timeout	is updated with the value in ms currently in use.
---------	---

Response

See Appendix A for a full list of status codes.

G4_CmdSetTimeout

This command changes the timeout value for the next commands. This changes the time needed by the reader to respond to the command. It varies according to the card in use.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

C

```
INT16 G_DECL G4_CmdSetTimeout
(
  const WORD16 ChannelNb,
  const WORD32 Timeout
)
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_CmdSetTimeout Lib "WGCR40.DLL" (ByVal ChannelNb
As Integer, ByVal Timeout As Long) As Integer
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_CmdSetTimeout Lib "W32GCR40.DLL" (ByVal
ChannelNb As Integer, ByVal Timeout As Long) As Integer
```

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
Timeout	holds the value to be set.

Response

See Appendix A for a full list of status codes.

G4_ICCDetection

This command installs a software check for the card detection.

Compatibility

Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C          INT16 G_DECL G4_ICCDetection
           (
           const WORD16 ChannelNb,
           const WORD16 Time,
           const INT16 (G_DECL *FnCheckIcc) (const INT32)
           const HINSTANCE hinst
           )
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_ICCDetection Lib "WGCR40.DLL" (ByVal ChannelNb
As Integer, ByVal Time As Integer, ByVal FnCheckIcc as Long, ByVal
hInst as Integer) As Integer
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_ICCDetection Lib "W32GCR40.DLL" (ByVal ChannelNb
As Integer, ByVal Time As Integer, ByVal FnCheckIcc as Long, ByVal
hInst as Integer) As Integer
```

Parameters In

ChannelNb	indicates the logical number allocated to the channel
Time	value in ms used by timer process (must be >=200ms). If value =0, detection mode is deactivated
FnCheckIcc	function to call when card is inserted or removed
hInst	takes the value of the instance of the application which holds the exported function

Response

See Appendix A for a full list of status codes.

G4_ICCStatus

This command reads the protocol currently in use between the reader and the card.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format**C**

```
INT16 G_DECL G4_ICCStatus
(
  const WORD16 ChannelNb,
  const WORD16 G_FAR *Protocol,
  WORD32 G_FAR *BaudRate,
  G4_SESSION_PARAM G_FAR *Session
)
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_ICCStatus Lib "WGCR40.DLL" (ByVal ChannelNb As Integer, Protocol As Integer, BaudRate As Long, Session As G4_SESSION_PARAM) As Integer
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_ICCStatus Lib "W32GCR40.DLL" (ByVal ChannelNb As Integer, Protocol As Integer, BaudRate As Long, Session As G4_SESSION_PARAM) As Integer
```

Parameters In

ChannelNb	indicates the logical number allocated to the channel
-----------	---

Parameters Out

Protocol	Card protocol, usually known as T=0, 1
BaudRate	Communication speed between card and reader
Session	Last received Answer to Reset

Response

See Appendix A for a full list of status codes.

G4_IFDDownload

This command enables data to be downloaded in an IFD. The input file “memory file” is the file to download in the GPR. The configuration file has the same name and path with a GPR extension. For example:

C:\GEMPLUS\GPR400\FILE1.BIN is the data file.

C:\GEMPLUS\GPR400\FILE1.GPR is the configuration file.

The configuration file is an ASCII file which contains parameters for the download operation. These parameters are as follows:

ADDRESS: 1C000 (address in hexadecimal notation)

DESTINATION: RAM (memory destination RAM or FLASH)

Note: As regard the FLASH memory, the function erases the memory before the download operation.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C
INT16 G_DECL G4_IFDDownload
(
const WORD16 ChannelNb,
const WORD8 G_FAR *MemoryFile
)
```

Visual Basic 3 for Windows 3.1

```
Declare Function G4_IFDDownload Lib "WGCR40.DLL" (ByVal ChannelNb As Integer, ByVal MemoryFile As String) As Integer
```

Visual Basic 4 for Windows 32 bits

```
Declare Function G4_IFDDownload Lib "W32GCR40.DLL" (ByVal ChannelNb As Integer, ByVal MemoryFile As String) As Integer
```

Parameters In

ChannelNb indicates the logical number allocated to the channel

MemoryFile holds the full file name for an IFD specific file to download

Response

See Appendix A for a full list of status codes.

G4_IFDStatus

This command returns information about the reader.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C
INT16 G_DECL G4_IFDStatus
(
  const WORD16 ChannelNb,
  G4_IFD_STATUS G_FAR *Status
)
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_IFDStatus Lib "WGCR40.DLL" (ByVal ChannelNb As Integer, Status As G4_IFD_STATUS) As Integer
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_IFDStatus Lib "W32GCR40.DLL" (ByVal ChannelNb As Integer, Status As G4_IFD_STATUS) As Integer
```

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
-----------	---

Parameters Out

Status.Parameters	stores the communication parameters used for the G4_OpenChannel command.
Status.Protocol	returns the protocol used between host and reader.
Status.ExchangeSize	holds the communication buffer size currently available in reader.
Status.OSString [MAX_IFD_STRING]	holds a null terminated string which identifies the OS name and version.
Status.SpecificLength	holds the number of updated byte in Specific buffer.
Status.Specific [MAX_IFD_SPECIFIC].	holds proprietary information for selected reader.

Response

If everything is OK: G_OK;
See Appendix for full list of status codes.

Remarks

The Parameters hold the communication parameters given to G4_OpenChannel (IFDType, IFDMode, Comm)

- The protocol returned is PCCARD_PROTOCOL
- ExchangeSize holds the communication buffer size available in the GPR400
- OSString holds a null terminated string identifying:
The OS name and version, format OS Version x.x
- SpecificLength holds the number of updated bytes in Specific buffer
- Specific holds the proprietary information detailed below:
 - Flash Memory present or not
 - ICC driver loaded or not
 - IC Card inserted or not
 - ICC Clock speed 3.68 or 7.36 Mhz

G4_IFDExchange

This command enables an exchange with the IFD. Refer to your reader reference manual for details on the download file format.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```

C      INT16 G_DECL G4_IFDExchange
      (
      const WORD16 ChannelNb,
      const WORD32 Timeout,
      const WORD32 SendLength,
      const WORD8 G_HUGE *SendBuffer,
      WORD32 G_FAR *ReadLength,
      WORD8 G_HUGE *ReadBuffer
      )

Visual Basic      Declare Function G4_IFDExchange Lib "WGCR40.DLL" (ByVal ChannelNb As
3 for              Integer, ByVal Timeout As Long,
Windows 3.1      ByVal SendLength As Long,
                  ByVal SendBuffer As String,
                  ReadLength As Long,
                  ByVal ReadBuffer As String) As Integer

Visual Basic      Declare Function G4_IFDExchange Lib "WGCR40.DLL" (ByVal ChannelNb As
4 for              Integer, ByVal Timeout As Long,
Windows 32      ByVal SendLength As Long,
bits             ByVal SendBuffer As String,
                  ReadLength As Long,
                  ByVal ReadBuffer As String) As Integer

```

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
SendLength	holds the number of bytes to be sent.
SendBuffer	holds the bytes to be sent. When you enter a reader command, the communication protocol is automatically added by this function.
ReadLength	holds the maximum number of data to be read. ReadBuffer must be able to receive ReadLength bytes.

Parameters Out

ReadLength	holds the number of bytes available in ReadBuffer.
ReadBuffer	holds the read bytes. It is a reader response without the protocol byte.

Response

See Appendix A for a full list of status codes.

Remarks

The GPR400 uses TLV commands described in the GPR400 Reference Manual.

G4_IFDGetPowerTimeout

This command returns the power timeout for the selected reader. This is the time needed by the IFD in order to remove power from the card when the G4_OpenSession command is used. The time default value is 100 ms.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format*C*

```
INT16 G_DECL G4_IFDGetPowerTimeout
(
  const WORD16 ChannelNb,
  const WORD32 G_FAR *Time,
)
```

*Visual Basic
3 for
Windows 3.1*

```
Declare Function G4_IFDGetPowerTimeout Lib "WGCR40.DLL" (ByVal
  ChannelNb As Integer, Timeout As Long) As Integer
```

```
)
```

*Visual Basic
4 for
Windows 32
bits*

```
Declare Function G4_IFDGetPowerTimeout Lib "W32GCR40.DLL" (ByVal
  ChannelNb As Integer, Timeout As Long) As Integer
```

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
-----------	---

Parameters Out

Time/Timeout	is updated with the value currently in use in ms
--------------	--

Response

See Appendix A for a full list of status codes.

G4_IFDSetPowerTimeout

This command changes the power timeout for the selected reader. This is the time needed by the reader in order to remove power from the card when the G4_OpenSession command is used.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C          INT16 G_DECL G4_IFDSetPowerTimeout
          (
          const WORD16 ChannelNb,
          const WORD32 Time,
          )
```

```
Visual Basic
3 for
Windows 3.1
          Declare Function G4_IFDSetPowerTimeout Lib "WGCR40.DLL" (ByVal
          ChannelNb As Integer, ByVal Timeout As Long) As Integer
```

```
Visual Basic
4 for
Windows.32
          Declare Function G4_IFDSetPowerTimeout Lib "W32GCR40.DLL" (ByVal
          ChannelNb As Integer, ByVal Timeout As Long) As Integer
```

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
Time/Timeout	is the value to be set in ms

Response

See Appendix A for a full list of status codes.

G4_ICCSetProtocol

Use this command to:

- change the communication protocol between the reader and the card (e.g. T0 to T1)
- change the communication speed between the reader and the card.

The file sent as a parameter contains the commands to be executed by the reader and the card for the protocol or the speed to be changed.

Note: Contact your sale representative to obtain any of these files.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```

C      INT16 G_DECL G4_ICCSetProtocol
      (
      const WORD16 ChannelNb,
      const char G_FAR *ProtocolFile,
      )

Visual Basic 3 for Windows 3.1
      Declare Function G4_ICCSetProtocol Lib "WGCR40.DLL" (ByVal ChannelNb
      As Integer, ByVal ProtocolFile As String) As Integer

Visual Basic 4 for Windows 32 bits
      Declare Function G4_ICCSetProtocol Lib "W32GCR40.DLL" (ByVal
      ChannelNb As Integer, ByVal ProtocolFile As String) As Integer

```

Parameters In

ChannelNb	indicates the logical number associated during open function to the selected channel.
ProtocolFile	holds a IFD specific file which contains data needed to initialize the target reader.

Response

See Appendix A for a full list of status codes.

GPR400 Specific Commands

G4_ODPowerDown

This command puts the reader in Power Down mode; this means that the reader is in a low energy mode. If no session has previously been opened, the reader will be in an even lower energy mode. The G4_ODResetGPR command will wake -Power- up the reader.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```

C      INT16 G_DECL G4_ODPowerDown
      (
          const WORD16 ChannelNb,
      )

Visual Basic 3 for  Declare Function G4_ODPowerDown Lib "WGR0D.DLL" (
Windows 3.1        ByVal ChannelNb As Integer,
                    ) As Integer

Visual Basic 4 for  Declare Function G4_ODPowerDown Lib "W32GR0D.DLL" (
Windows 32 bits    ByVal ChannelNb As Integer,
                    ) As Integer
  
```

Parameters In

ChannelNb Holds the communication channel Number.

Response

See Appendix A for a full list of status codes

G4_0DStandBy

This command puts the reader in Standby mode. Either a card insertion or the G4_0DReset GPR command can wake up the reader.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```

C          INT16 G_DECL G4_0DStandBy
          (
            const WORD16 ChannelNb,
          )

Visual Basic 3 for Declare Function G4_0DStandBy Lib "WGR0D.DLL" (
Windows 3.1      ByVal ChannelNb As Integer,
                  ) As Integer

Visual Basic 4 for Declare Function G4_0DStandBy Lib "W32GR0D.DLL" (
Windows 32 bits  ByVal ChannelNb As Integer,
                  ) As Integer

```

Parameters In

ChannelNb Holds the communication channel Number.

Response

See Appendix A for a full list of status codes

G4_ODResetGPR

This command resets the GPR400. The RAM is then erased.

If the reader is in Power Down mode (G4_ODPowerDown) or in Stand By mode (G4_ODStandBy) the G4_ODResetGPR command wakes the reader up. The content of the RAM remains unchanged.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```

C          INT16 G_DECL G4_ODResetGPR
          (
            const WORD16 ChannelNb,
          )

Visual Basic 3 for Windows 3.1  Declare Function G4_ODResetGPR Lib "WGR0D.DLL" (
                                ByVal ChannelNb As Integer,
                                ) As Integer

Visual Basic 4 for Windows 32 bits  Declare Function G4_ODResetGPR Lib "W32GR0D.DLL" (
                                ByVal ChannelNb As Integer,
                                ) As Integer

```

Parameters In

ChannelNb Holds the communication channel Number.

Response

See Appendix A for a full list of status codes.

G4_ODSwitchClock

This command switches the GPR400 clock frequency from 3.68 Mhz to 7.36 MHz or from 7.36 MHz to 3.68 Mhz. This function is only available for microprocessors cards.

Warning!: Check that all cards functions - especially write operations- are operational at 7mhz.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C      INT16 G_DECL G4_ODSwitchClock
      (
        const WORD16 ChannelNb,
        WORD16 G_FAR *ClockGPR
      )
```

```
Visual Basic 3 for  Declare Function G4_ODSwitchClock Lib "WGR0D.DLL" (
Windows 3.1      ByVal ChannelNb As Integer,
                  ClockGPR As Integer
                  ) As Integer
```

```
Visual Basic 4 for  Declare Function G4_ODSwitchClock Lib "W32GR0D.DLL" (
Windows 32 bits   ByVal ChannelNb As Integer,
                  ClockGPR As Integer
                  ) As Integer
```

Parameters In

ChannelNb : Holds the communication channel Number.

Parameters Out

ClockGPR : Contains the new clock frequency (368 for 3.68MH or 736 for 7.36 MH) if the response is G_OK.

Response

See Appendix A for a full list of status codes.

G4_ODLoadMemory

This command loads data in GPR400 memory (RAM or Flash).

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```

C      INT16 G_DECL G4_ODLoadMemory
      (
        const WORD16      ChannelNb,
        const WORD16      Destination,
        const WORD16      DeleteBefore,
        const WORD32      StartAddr,
        const BYTE        G_FAR *CardPath
      );

```

Visual Basic 3 for Windows 3.1

```

      Declare Function G4_ODLoadMemory Lib "WGR0D.DLL" (
        ByVal ChannelNb As Integer,
        ByVal Destination As Integer,
        ByVal DeleteBefore As Integer,
        ByVal StartAddr As Long,
        ByVal CardPath As String
      ) As Integer

```

Visual Basic 4 for Windows 32 bits

```

      Declare Function G4_ODLoadMemory Lib "W32GR0D.DLL" (
        ByVal ChannelNb As Integer,
        ByVal Destination As Integer,
        ByVal DeleteBefore As Integer,
        ByVal StartAddr As Long,
        ByVal CardPath As String
      ) As Integer

```

Parameters In

ChannelNb	Holds the communication channel Number.
Destination	0 is for RAM 1 is for Flash
DeleteBefore	0 Do not delete code before 1 Delete code before Used only for Flash memory
StartAddr	Holds the base start address where the data must be loaded: Range: 0 to 0x6D4 for RAM loading, depending on the length of data to load 0 to 0xFFFF for Flash loading
CardPath	Pathname to be loaded in the GPR400 memory

Response

See Appendix A for a full list of status codes.

G4_ODReadMemory

This command reads data in the GPR400 memory (RAM or Flash).

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C      INT16 G_DECL G4_ODReadMemory
      (
          const WORD16      ChannelNb,
          const WORD16      Location,
          const WORD32      StartAddr,
          const WORD32      Length,
          BYTE G_HUGE Data[]
      )
```

Visual Basic 3 for Windows 3.1

```
Declare Function G4_ODReadMemory Lib "WGR0D.DLL" (
    ByVal ChannelNb As Integer,
    ByVal Location As Integer,
    ByVal StartAddr As Long,
    ByVal Length As Long,
    ByVal Data As String
) As Integer
```

Visual Basic 4 for Windows 32 bits

```
Declare Function G4_ODReadMemory Lib "W32GR0D.DLL" (
    ByVal ChannelNb As Integer,
    ByVal Location As Integer,
    ByVal StartAddr As Long,
    ByVal Length As Long,
    ByVal Data As String
) As Integer
```

Parameters In

ChannelNb	Holds the communication channel Number.
Location	0 is for RAM 1 is for Flash.
StartAddr	is the first address where the function begin to read: Range: 0 to 0x6D4 for RAM loading 0 to 0x1FFFF for Flash loading
Length	Length of data to read.

Parameters Out

Data[]	Read data
--------	-----------

Response

See Appendix A for a full list of status codes.

G4_0DActivateDriver

This command is used to activate a Card driver or an OS filter previously loaded in RAM or FLASH memory with the G4_0DLoadmemory or G4_0DIFDDownload functions.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C          INT16 G_DECL G4_ActivateDriver
          (
            const WORD16 ChannelNb,
            const WORD16 DriverType,
            const WORD16 Location,
            const WORD32 StartAddr,
          )
```

```
Visual Basic 3 for Declare Function G4_0DActivateDriver Lib "WGR0D.DLL" (
Windows 3.1      ByVal ChannelNb As Integer,
                  ByVal DriverType As Integer,
                  ByVal Location As Integer,
                  ByVal StartAddr As Long,
                  ) As Integer
```

```
Visual Basic 4 for Declare Function G4_0DActivateDriver Lib "W32GR0D.DLL" (
Windows 32 bits  ByVal ChannelNb As Integer,
                  ByVal DriverType As Integer,
                  ByVal Location As Integer,
                  ByVal StartAddr As Long,
                  ) As Integer
```

Parameters In

ChannelNb	Holds the communication channel Number.
DriverType	0 is for an IC Card Driver 1 is for a OS filter
Location	0 is for RAM 1 is for Flash
StartAddr	Address where the driver is loaded

Response

See Appendix A for a full list of status codes.

Note: ICC Carddriver or OS filter should be developed by GEMPLUS. For more information see the GEMPLUS web (<http://www.gemplus.com/gpr400.html>).

G4_ODDeactivateDriver

This command is used to deactivate an IC Card driver or an OS filter previously activated by the G4_ODActivateDriver.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C          INT16 G_DECL G4_DeactivateDriver
          (
            const WORD16 ChannelNb,
            const WORD16 DriverType,
            const WORD16 Location,
            const WORD32 StartAddr,
          )
```

```
Visual Basic 3 for  Declare Function G4_ODDeactivateDriver Lib "WGR0D.DLL" (
Windows 3.1        ByVal ChannelNb As Integer,
                    ByVal DriverType As Integer,
                    ByVal Location As Integer,
                    ByVal StartAddr As Long,
                    ) As Integer
```

```
Visual Basic 4 for  Declare Function G4_ODDeactivateDriver Lib "W32GR0D.DLL" (
Windows 32 bits    ByVal ChannelNb As Integer,
                    ByVal DriverType As Integer,
                    ByVal Location As Integer,
                    ByVal StartAddr As Long,
                    ) As Integer
```

Parameters In

ChannelNb	Holds the communication channel Number.
DriverType	0 is for an IC Card Driver 1 is for a OS filter
Location	0 is for RAM 1 is for Flash
StartAddr	Address where the driver is loaded

Response

See Appendix A for a full list of status codes

G4_0DExecMemory

This command calls a loaded code. Other functions developed by GEMPLUS - for example specific functions for custom ICC protocol-, could be added in the GPR400 reader. For more information contact GEMPLUS.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C          INT16 G_DECL G4_0DExecMemory
          (
            const WORD16 ChannelNb,
            const BYTE G_FAR Sbuff[ ],
            const WORD16 Lin,
            BYTE G_FAR Rbuff[ ],
            WORD16 G_FAR *Lout,
          )
```

```
Visual Basic 3 for Declare Function G4_0DExecMemory Lib "WGR0D.DLL" (
Windows 3.1      ByVal ChannelNb As Integer,
                  ByVal Sbuf as String,
                  ByVal Lin As Integer
                  ) As Integer
```

```
Visual Basic 4 for Declare Function G4_0DExecMemory Lib "W32GR0D.DLL" (
Windows 32 bits  ByVal ChannelNb As Integer,
                  ByVal Sbuf as String,
                  ByVal Lin As Integer
                  ) As Integer
```

Parameters In

ChannelNb	Holds the communication channel Number.
Lin	Holds the length of data to send to GPR400
SBuff	Holds data to send to GPR400

Parameters Out

Rbuff	Holds filter return data
Lo	Holds length of filter return data

Response

See Appendix A for a full list of status codes.

G4_0DGetCIS

This command returns a string which contains information about the PC Card Driver and the GPR400 inserted in the specified socket.

Compatibility

DOS, Windows 3.1, Windows 95, Windows NT Version 4.

Format

```
C          INT16 G_DECL G4_0DGetCIS
          (
            const WORD16      ChannelNb,
              BYTE   G_FAR   *Length,
              BYTE   G_FAR   Data[]
          )
```

**Visual Basic 3
for Windows 3.1**

```
Declare Function G4_0DGetCIS Lib "WGR0D.DLL" (
  ByVal ChannelNb As Integer,
  Le As Integer,
  ByVal InfoString As String
) As Integer
```

**Visual Basic
4 for
Windows 32
bits**

```
Declare Function G4_0DGetCIS Lib "W32GR0D.DLL" (
  ByVal ChannelNb As Integer,
  Le As Integer,
  ByVal InfoString As String
) As Integer
```

Parameters In

ChannelNb Holds the communication channel Number.

Parameters Out

Length Length of data returned by the function;
typical length: 1024 bytes

Data Card Services info string. Data is actually a pointer to the
memory area allocated to receive data created by the
function.

Response

See Appendix A for a full list of status codes.

VISUAL BASIC UTILITIES

The GCR Interface Library provides the following specific functions that optimize the compatibility of the GCR Interface Library commands and structures with Visual Basic. These functions are found in VBGTOOL4.DLL for Windows 3.1 and VBTOOL32.DLL for Windows .

Visual Basic Utilities for Windows

G4_StrToPtr

This function converts a Visual Basic string to a Long variable that is the address of the string. It must be used with the G4_ExchangeA pdu command to allocate the buffers.

Compatibility

Windows 3.1.

Format

*Visual Basic 3
for Windows 3.1*

```
Declare Sub G4_StrToPtr Lib "VBGTOOL4.DLL" ( ByVal VarString As
String) As Long
```

Parameters In

Enter the following parameters with this command.

Parameter	Definition
VarString	Visual Basic variable-length string

Response

This function returns the address of the variable-length string.

G4_ByteArrayToPtr

This function converts a Visual Basic array of bytes to a Long variable that is the address of the array of bytes. It must be used with the G4_ExchangeApu command to allocate the buffers.

Compatibility

Windows 95, Windows NT Version 4.

Format

*Visual Basic 4
for Windows 32
bits*

```
Declare Sub G4_ByteArrayToPtr Lib "VBTOOL32.DLL" ( ByFirstByte As  
Byte) As Long
```

Parameters In

Enter the following parameters with this command.

Parameter	Definition
FirstByte	First byte of the array

Response

This function returns the address of the array of bytes.

G4_WritePCCARDComm

This procedure updates the Comm union fields of a G4_CHANNEL_PARAM structure with PCCARD communication parameters.

See *GCR Interface Library Programmer's Guide-Generic Commands* for detail of the G4_CHANNEL_PARAM structure.

Compatibility

Windows 3.1, Windows 95, Windows NT Version 4.

Format

*Visual Basic 3
for Windows 3.1*

```
Declare Sub G4_WritePCCARDComm Lib "VBGTOOL4.DLL" (Channel Param As
G4_CHANNEL_PARAM, ByVal Socket As Integer, ByVal ITNumber As
Integer)
```

*Visual Basic 4
for Windows 32
bits*

```
Declare Sub G4_WritePCCARDComm Lib "VBTOOL32.DLL" (Channel Param As
G4_CHANNEL_PARAM, ByVal Socket As Integer, ByVal ITNumber As
Integer)
```

Parameters In

Enter the following parameters with this command.

ChannelParam	A G4_CHANNEL_PARAM whose union must be updated
Socket	PCCard identifier
ITNumber	Interrupt associated with the selected port

Parameters Out

ChannelParam	The Comm union is updated with the input values. The IFDMode field is not updated
--------------	---

G4_ReadPCCARDComm

This procedure reads the Comm union fields of a G4_CHANNEL_PARAM structure assuming that the communication protocol is PCCard.

See above for detail of the G4_CHANNEL_PARAM structure.

Compatibility

Windows 3.1, Windows 95, Windows NT Version 4.

Format

*Visual Basic 3
for Windows 3.1*

```
Declare Sub G4_ReadPCCARDComm Lib "VBGT00L4.DLL" (ChannelParam As  
G4_CHANNEL_PARAM, Socket As Integer, ITNumber As Integer)
```

*Visual Basic 4
for Windows 32
bits*

```
Declare Sub G4_ReadPCCARDComm Lib "VBT00L32.DLL" (ChannelParam As  
G4_CHANNEL_PARAM, Socket As Integer, ITNumber As Integer)
```

Parameters In

Enter the following parameters with this command.

ChannelParam	A G4_CHANNEL_PARAM whose IFDMode is presumed to be G_PC_CARD.
--------------	---

Parameters Out

Socket	PCCard identifier
ITNumber	Interrupt associated with the selected port

APPENDIX A. STATUS CODES

This appendix lists the status codes that may be returned by the GCR Interface Library, and describes their meanings. These may be one of the following types of status code.

- Host status codes
- Host/Reader status codes
- Reader status codes
- Reader/Card status codes
- Card status codes
- APDU status codes
- These are described below.

Host Status Codes

Host Status Codes are returned when a problem is encountered within the PC host.

For example, the specified port is not open, or a Windows error has occurred.

The host status codes are listed below.

Code	Meaning
-400	Specified port out of range
-401	Specified port non existent
-402	Specified port not initializes - specified parameters not correct
-403	Specified port in use
-404	Port break
-410	Communication problem under Windows 3.x
-411	Specified port is already open
-412	Specified port is already closed
-420	Memory allocation failure
-421	Bad pointer. Function unknown in DLL
-422	Host buffer too small
-430	Host has run out of resources
-440	Operation canceled by user
-450	Parameter out of allowed range
-451	Call failed. Target library not found
-452	Call failed. Target library does not implement this function.

Host/Reader Status Codes

Host/Reader Status Codes are returned when a problem is encountered between the host PC and the reader. For example, there was a communication error between the reader and the host PC, or a message length was too long.

The host/reader status codes are listed below.

Code	Meaning
-300	Communication error between host and reader
-301	Character parity error between reader and host
-302	Longitudinal redundancy code error
-310	Host-reader protocol frame error
-311	Incorrect LN parameter value in header
-312	Header does not contain ACK or NACK
-313	Message length sent from host not supported
-314	Reader returning NACK or R-block
-315	S-block returned by reader
-316	Incorrect source/target address
-317	Incorrect sequence number

Reader Status Codes

Reader Status Codes are returned when a problem is encountered with the reader itself. For example, the reader is poorly connected, or a requested command cannot be performed by the reader.

The Reader status codes are listed below.

Code	Meaning
-200	No reader connected
-201	No response from reader
-202	Reader is not supported
-210	Insufficient power for programming
-211	Command is not available for specified reader
-212	Command not executed. Reader timeout or incoherence in the argument number or type in a reader command
-213	A macro definition generated an internal error
-214	The called IFD function has failed
-215	This memory option is not available
-216	The selected memory access is forbidden
-217	The selected code cannot be activated
-220	Reader sent an abort block. Buffer too small to hold requested data from card.
-221	Reader has executed a resynchronisation and lost data
-290	Reader by the reader keyboard when no key has been pressed during the given time
-291	Returned by the reader keyboard when two keys have been pressed at the same time.

Reader/Card Status Codes

Reader/Card Status Codes are returned when a problem is encountered between the reader and the card. For example, the answer to reset is incoherent or an incorrect APDU command has been sent to the card.

The reader/card status codes are listed below.

Code	Meaning
-100	Communication is not possible between the card and the reader
-101	Character parity error between the card and the reader
-102	Error detection code raised
-110	Incoherence in ATR
-111	Incorrect TS in ATR
-112	Incorrect TCK in ATR
-113	Impossible to read some ATR bytes
-120	Inconsistent protocol
-121	Protocol unknown
-122	PTS is required
-123	Received block length is greater than IFSD
-124	Incorrect procedure byte received from card
-125	Incorrect INS entered in a command (6X or 9X)
-126	Message length from card not supported
-127	3 failures. Resynchronisation required by card

Card Status Codes

Card Status Codes are returned when a problem is encountered with the card. For example, there is no card present or the card was removed.

The card status codes are listed below.

Code	Meaning
-1	No card in a reader
-2	No response from card
-3	Card not supported
-4	Card was removed during command execution. The database may be corrupted
-5	Card not receiving power or it has been removed between two commands
-6	Card has short circuited in the reader or it is physically incompatible with the reader
-10	Card returned an abort block

APDU Status Codes

The following status codes are returned by the command as a result of an APDU related error.

Code	Meaning
-501	Channel already open
-502	Channel already closed
-503	No session opened with the card
-504	Switch session not available
-511	APDULenMax exceeds 65544
-512	Length must be less than 65536 in an APDU command
-513	The response must contain SW1 and SW2
-520	The selected reader module is not present in the system
-521	The reader module does not support the selected function

TLV Status Codes

The following status codes may be returned by a TLV related command.

Code	Meaning
-601	TLV type unknown
-602	Buffer too small to hold data relating to TLV type
-603	Requested operation not executed

File Management Status Codes

The following status codes may occur during file operations.

Code	Meaning
-700	File already opened or impossible to open. Requested DLL not installed
-701	Unable to close a file
-710	Unable to write a file
-720	Unable to read a file
-730	Invalid file format
-731	No header found in file
-732	Unmatched quotation mark found
-733	Unexpected end of file
-734	Invalid CRC value
-740	File version not supported
-750	Invalid configuration file
-800	Wait for object has failed
-801	API cannot release the semaphore

Warnings

1	The card is not fully supported
2	APDU cannot be transported with this value
3	Length= response length

Other Errors

-1xxx	Status codes below - 1000 reflects specific reader status code. xxx is the status code returned by the reader
-------	---

APPENDIX B. APDU COMMAND FORMAT

This version of the GPR400 Interface Library works with T=0 and T=1 protocol cards in such a way that it is transparent to the application. This is due to the use of the Application Protocol Data Unit (APDU) structure specified by the ISO 7816-4 standard.

An APDU is like a template, into which the various components of a command and its response are fitted. A command is sent from a reader to a card in one APDU, and its response is returned in the opposite direction in another.

Command APDU

Command APDUs consist of the following:

- a mandatory 4-byte header
- an optional body of parameter length

They are structured as follows:

Header				Body		
CLA	INS	P1	P2	Lc	Parameters/data	Le

The command APDU contents are described in the following table.

Command APDU Contents

Field Name	Length (in bytes)	Description
CLA	1	Instruction class
INS	1	Instruction code
P1	1	Parameter 1
P2	1	Parameter 2
Lc	1	Number of bytes of data that will be sent with the command (see data entry below)
Data	Lc	Command parameters or data
Le	1	Expected length of data to be returned

The contents of the CLA, INS, P1, and P2 fields depend on the command. See the appropriate card documentation for further details.

Response APDU

Response APDUs consist of the following:

- an optional body that stores the data returned by the card
- a mandatory trailer that stores the SW1, SW2 status bytes.

They are structured as follows:

Body	Trailer
Data	SW1, SW2

APDU Types

The GPR400 Interface Library supports the following ISO 7816-4 APDU types:

Case 1	No command or response data. (transported as T=0 ISO IN TPDU or T=1 TPDU with the length = 0)
Case 2S	No command data but with between 1 and 256 bytes response data. (transported as T=0 ISO OUT TPDU or T=1 TPDU)
Case 3S	Command data between 1 and 255 bytes and no response data. (transported as T=0 ISO IN TPDU)
Case 4S	Command data between 1 and 255 bytes, response data between 1 and 256 bytes. (transported as a T=0 ISO IN TPDU or T=1 TPDU. For T=0 TPDU, the command must be followed by a <i>i_GetResp</i> command transported as a T=0 ISO OUT TPDU. The <i>i_GetResp</i> mechanism conforms to the ISO 7816-4 standard)

APPENDIX C.

In order to obtain:

- versions dedicated to specific platforms
- ICC Card Drivers for memory cards or custom cards
- any API update

Connect yourself to the GEMPLUS page on the Word Wide Web at the following address:

<http://www.gemplus.com./gpr400.html>