

GSEOS Language Runtime Library Description

Revision 1.12

IDA-GSEOS-0004

May 2004

Prepared by
Kai Stoeckner

Tables and Figures	8
1 Scope	9
1.1 Purpose of this Document	9
1.2 Change Record	9
1.3 Reference Documentation	10
1.4 Abbreviations	11
2 Introduction	12
3 Common #defines	13
3.1 Type #defines	13
3.2 Other #defines	13
4 RTL Blocks	14
4.1 Block _Timer1Sec	15
4.2 Block _Serial	16
4.3 Block _Net	17
4.4 Block _NetNotification	18
4.5 Block _Message	19
5 Min/Max Functions	20
6 Command Channel Functions	21
6.1 _SendCommand ()	22
7 Dialog Input Functions	23
7.1 _DlgGetChar ()	24
7.2 _DlgGetCharArray ()	25
7.3 _DlgGetDouble ()	26
7.4 _DlgGetDoubleArray ()	27
7.5 _DlgGetI64 ()	28
7.6 _DlgGetI64Array ()	29
7.7 _DlgGetLong ()	30
7.8 _DlgGetLongArray ()	31
7.9 _DlgGetOpenFileName ()	32
7.10 _DlgGetSaveFileName ()	33
7.11 _DlgGetShort ()	34
7.12 _DlgGetShortArray ()	35
7.13 _DlgGetString ()	36
7.14 _DlgGetUChar ()	37

7.15	_DlgGetUCharArray ()	38
7.16	_DlgGetui64 ()	39
7.17	_DlgGetui64Array ()	40
7.18	_DlgGetULong ()	41
7.19	_DlgGetULongArray ()	42
7.20	_DlgGetUShort ()	43
7.21	_DlgGetUShortArray ()	44
8	Network Functions	45
8.1	_NetClose ()	46
8.2	_NetConfig ()	47
8.3	_NetIsOpen ()	49
8.4	_NetOpen ()	50
8.5	NetSend ()	51
9	Serial Port Functions	52
9.1	_SerialClose ()	53
9.2	_SerialConfig ()	54
9.3	_SerialIsOpen ()	55
9.4	_SerialEscapeFunction ()	56
9.5	_SerialOpen ()	57
9.6	SerialSend ()	58
10	File Functions	59
10.1	_FileClose ()	61
10.2	_FileConfig ()	62
10.3	_FileIsOpen ()	63
10.4	_FileOpen ()	64
10.5	_FileRead ()	65
10.6	_FileReadAscIntoCharArray ()	66
10.7	_FileReadAscIntoDoubleArray ()	67
10.8	_FileReadAscIntoi64Array ()	68
10.9	_FileReadAscIntoLongArray ()	69
10.10	_FileReadAscIntoShortArray ()	70
10.11	_FileReadAscIntoUCharArray ()	71
10.12	_FileReadAscIntoui64Array ()	72
10.13	_FileReadAscIntoULongArray ()	73

10.14	<code>_FileReadAscIntoUShortArray ()</code>	74
10.15	<code>_FileSeek ()</code>	75
10.16	<code>FileWrite ()</code>	76
11	Converting Functions	77
11.1	<code>_AscToCharArray ()</code>	78
11.2	<code>_AscToDoubleArray ()</code>	79
11.3	<code>_AscToLongArray ()</code>	80
11.4	<code>_AscToShortArray ()</code>	81
11.5	<code>_AscToUCharArray ()</code>	82
11.6	<code>_AscToULongArray ()</code>	83
11.7	<code>_AscToUShortArray ()</code>	84
11.8	<code>_atod ()</code>	85
11.9	<code>_atoi64 ()</code>	86
11.10	<code>_atol ()</code>	87
11.11	<code>_atoui64 ()</code>	88
11.12	<code>_atoul ()</code>	89
11.13	<code>_dtoa ()</code>	90
11.14	<code>_dtof32 ()</code>	91
11.15	<code>_dtof32Array ()</code>	92
11.16	<code>_dtoui64 ()</code>	93
11.17	<code>_f32tod ()</code>	94
11.18	<code>_f32todArray ()</code>	95
11.19	<code>_Gray16ToRgb32 ()</code>	96
11.20	<code>_GrayS16ToRgb32 ()</code>	97
11.21	<code>_i64toa ()</code>	98
11.22	<code>_i64tod ()</code>	99
11.23	<code>_ltoa ()</code>	100
11.24	<code>_sprintf ()</code>	101
11.25	<code>_ToIntel16 ()</code>	105
11.26	<code>_ToIntel32 ()</code>	106
11.27	<code>_ToIntel64 ()</code>	107
11.28	<code>_ToIntel16Array ()</code>	108
11.29	<code>_ToIntel32Array ()</code>	109
11.30	<code>_ToIntel64Array ()</code>	110

11.31	_ToMotorola16 ()	111
11.32	_ToMotorola32 ()	112
11.33	_ToMotorola64 ()	113
11.34	_ToMotorola16Array ()	114
11.35	_ToMotorola32Array ()	115
11.36	_ToMotorola64Array ()	116
11.37	_ui64toa ()	117
11.38	_ultoa ()	118
12	Time Functions	119
12.1	_GetSystemTime ()	120
12.2	_GetSystemTimei64 ()	121
12.3	_GetUnixTime ()	122
12.4	_i64SystemTimeToString ()	123
12.5	_i64ToSystemTime ()	124
12.6	_LocalTimeToSystemTime ()	125
12.7	_RPROGetCurrentTime ()	126
12.8	_RPROSetCurrentTime ()	127
12.9	_SetSystemTime ()	128
12.10	_SetSystemTimei64 ()	129
12.11	_SystemTimeToi64 ()	130
12.12	_SystemTimeToLocalTime ()	131
12.13	_SystemTimeToString ()	132
12.14	_SystemTimeToUnixTime ()	134
12.15	_UnixTimeToSystemTime ()	135
13	String Functions	136
13.1	_memcmp ()	137
13.2	_memmove ()	138
13.3	_memset ()	139
13.4	_strcat ()	140
13.5	_strcmp ()	141
13.6	_strcmpi ()	142
13.7	_strcpy ()	143
13.8	_strlen ()	144
13.9	_strlwr ()	145

13.10	<code>_strncat ()</code>	146
13.11	<code>_strncmp ()</code>	147
13.12	<code>_strncmpi ()</code>	148
13.13	<code>_strncpy ()</code>	149
13.14	<code>_strnset ()</code>	150
13.15	<code>_strrev ()</code>	151
13.16	<code>_strset ()</code>	152
13.17	<code>_strupr ()</code>	153
14	64-Bit Integer Functions	154
14.1	<code>_i64Add ()</code>	155
14.2	<code>_i64Cmp ()</code>	156
14.3	<code>_i64Div ()</code>	157
14.4	<code>_i64Mul ()</code>	158
14.5	<code>_i64Shra ()</code>	159
14.6	<code>_i64Sub ()</code>	160
14.7	<code>_ui64Add ()</code>	161
14.8	<code>_ui64And ()</code>	162
14.9	<code>_ui64Cmp ()</code>	163
14.10	<code>_ui64Div ()</code>	164
14.11	<code>_ui64Mul ()</code>	165
14.12	<code>_ui64Not ()</code>	166
14.13	<code>_ui64Or ()</code>	167
14.14	<code>_ui64Rol ()</code>	168
14.15	<code>_ui64Ror ()</code>	169
14.16	<code>_ui64Shl ()</code>	170
14.17	<code>_ui64Shrl ()</code>	171
14.18	<code>_ui64Sub ()</code>	172
14.19	<code>_ui64Xor ()</code>	173
15	Multimedia Functions	174
15.1	<code>_MciPlay ()</code>	175
15.2	<code>_MciSendString ()</code>	176
16	Miscellaneous Functions	177
16.1	<code>_AddMessage ()</code>	178
16.2	<code>_Beep ()</code>	179

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 7 of: 187
Project: GSEOS		

16.3	_GetBdmQueueFillState ()	180
16.4	_GetBlockId ()	181
16.5	_GetQueueDriverHandle ()	182
16.6	_MessageBox ()	183
16.7	_StartApplication ()	185
16.8	_SetThreadPriority ()	186
16.9	_SyncIdx ()	187

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 8 of: 187
Project: GSEOS		

Tables and Figures

Table 1 Change Record	9
Table 2 Abbreviations	11
Table 3 Command Channels.....	21
Table 4 Mode String Values.....	59

1 Scope

1.1 Purpose of this Document

Purpose of this document is to describe the GSEOS Language Runtime Library Functions. The Runtime Library Functions are predefined functions builds in the GSEOS.

1.2 Change Record

Table 1 Change Record

Date	Revision	Author	Affected Sections
6/4/1999	1.0	Stoekner	All sections
6/7/1999	1.1	Stoekner	0
2/7/2000	1.2	Stoekner	1.4, 3.1, 4.3, 4.4, 6.1, 8-8.5, 9, 9.5, 9.6, 10.4, 10.15, 11.8, 16.7
2/18/2000	1.3	Stoekner	9, section <code>_SerialFlush</code> deleted
6/5/2000	1.4	Stoekner	major modifications, several sections added: Min/Max Functions Time Functions String Functions 64-Bit Integer Functions several sections enhanced: File Functions Converting Functions
8/1/2000	1.5	Stoekner	modified: 12.4, 12.12, 10.15 functions added: <code>_ui64And</code> , <code>_ui64Not</code> , <code>_ui64Or</code> , <code>_ui64Xor</code> , <code>_ToIntel64</code> , <code>_ToMotorola64</code> , <code>_memset</code> , <code>_SerialEscapeFunction</code>
8/16/2000	1.6	Stoekner	4.3, 4.4
8/18/2000	1.7	Stoekner	12.6, 12.8, 16.3
2/15/2001	1.8	Stoekner	modified: 10 functions added: <code>_StartApplication</code> , <code>_MciPlay</code> , <code>_MciSendString</code> , <code>_ToIntel16Array</code> , <code>_ToIntel32Array</code> , <code>_ToIntel64Array</code> , <code>_ToMotorola16Array</code> , <code>_ToMotorola32Array</code> , <code>_ToMotorola64Array</code> ,

Date	Revision	Author	Affected Sections
			<code>_dtof32, _dtof32Array, _f32tod, _f32todArray</code>
6/29/2001	1.9	Stoekner	modified: 6.1, functions renamed: (defines for compatibility) <code>_DlgGetByte</code> → <code>_DlgGetChar</code> <code>_DlgGetByteArray</code> → <code>_DlgGetCharArray</code> <code>_DlgGetUByte</code> → <code>_DlgGetUChar</code> <code>_DlgGetUByteArray</code> → <code>_DlgGetUCharArray</code> <code>_FileReadAscIntoByteArray</code> → <code>_FileReadAscIntoCharArray</code> <code>_FileReadAscIntoUByteArray</code> → <code>_FileReadAscIntoUCharArray</code> <code>_AscToByteArray</code> → <code>_AscToCharArray</code> <code>_AscToUByteArray</code> → <code>_AscToUCharArray</code> functions added: <code>_SystemTimeToLocalTime</code> , <code>_LocalTimeToSystemTime</code> , <code>_DlgGeti64</code> , <code>_DlgGeti64Array</code> , <code>_DlgGetui64</code> , <code>_DlgGetui64Array</code> , <code>_FileReadAscIntoi64Array</code> , <code>_FileReadAscIntoui64Array</code>
8/15/2001	1.10	Stoekner	added: 4.5 Block <code>_Message</code> , 11.24 <code>_sprintf ()</code>
6/18/2002	1.11	Stoekner	modified: 1.4, 4.3, 4.4, 5, 8, 8.2, 8.4 added: 13.2 <code>_memmove ()</code>
5/25/2004	1.12	Stoekner	modified: 10.8, 10.12 functions added: 16.8 <code>_SetThreadPriority()</code> , 16.4 <code>_GetBlockId ()</code> , 16.5 <code>_GetQueueDriverHandle()</code>

1.3 Reference Documentation

All GSEOS related documents are listed in the “GSEOS Document Index” (IDA-GSEOS-0000).

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 11 of: 187
Project: GSEOS		

1.4 Abbreviations

Table 2 Abbreviations

EGSE	<u>E</u> lectrical <u>G</u> round <u>S</u> upport <u>E</u> quipment
GSEOS	<u>G</u> round <u>S</u> upport <u>E</u> quipment <u>O</u> perating <u>S</u> ystem
MGSE	<u>M</u> echanical <u>G</u> round <u>S</u> upport <u>E</u> quipment
RPRO	<u>R</u> osetta <u>C</u> ommon <u>P</u> ack- <u>e</u> tised <u>P</u> rotocol

RTL	<u>R</u> untime <u>L</u> ibrary
TCP/IP	<u>T</u> ransmission <u>C</u> ontrol <u>P</u> rotocol/ <u>I</u> nternet <u>P</u> rotocol
UDP	<u>U</u> ser <u>D</u> atagram <u>P</u> rotocol

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1 style="margin: 0;">GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 12 of: 187
Project: GSEOS		

2 Introduction

The GSEOS Language is the common configuration language for GSEOS. The language is mostly C-like. A full description of the language is given in the document “GSEOS Language Description” (IDA-GSEOS-0003).

Like the original C-language, the GSEOS Language has only low level statements. All high level functions are in a so-called Runtime Library. The language is able to call such external functions.

The Runtime Library functions of the GSEOS Language are coded in plain C++. To make them accessible by the GSEOS-Language, all function prototypes, blocks, and structures are defined in the file **gseos.g**. The file should be included at the beginning of the compiler main file. A simple main file is shown below:

```
#include "gseos.g"

void _main ()
{
  _MessageBox ("Test", "Starting!", MB_OK | MB_ICONINFORMATION);
}

void _exit ()
{
}

decode on (_Timer1Sec)
{
  _Beep ();
}
```

There are two special functions **_main ()** and **_exit ()**. The function **_main ()** is the first function called by the GSEOS on startup just before the source switch is set to on and blocks can come into the system. The function **_exit ()** is called on GSEOS shutdown just after the source switch is set to off.

All Runtime Library functions, system generated blocks, and structures used by the system have names beginning with an underscore (“_”). So the user can simply identify the predefined functions, blocks, and structures.

The user should never change names beginning with an underscore or any other part of the file **gseos.g**. If the function prototype definitions does not fit to the library functions builds in the GSEOS the program may crash.

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG Project: GSEOS	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 13 of: 187
--	---	---

3 Common #defines

This section describes the #defines in the file **gseos.g**.

3.1 Type #defines

The following general types are defined.

```
#define ULONG    unsigned long
#define LONG     long
#define USHORT   unsigned short
#define SHORT    short
#define UCHAR    unsigned char
#define CHAR     char
#define BOOL     unsigned long
#define VOID     void
#define DOUBLE   double
#define FLOAT    double
#define BITMAP   bitmap
#define VOID     void
#define CONST    const
```

3.2 Other #defines

There are some other useful #defines.

```
#define TRUE     1
#define FALSE    0
#define MAX_PATH 260 // Windows NT max. path length
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 14 of: 187

4 RTL Blocks

This section describes the various system-generated blocks. The user should not change the names and the structure of the blocks. The usual way to use a system-generated block is to write a decoder attached to the block.

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 15 of: 187
Project: GSEOS		

4.1 Block `_Timer1Sec`

Description The system sends the block `_Timer1Sec` every one second. The block may be used for polling purposes.

Declaration

```
typedef struct
{
    ULONG ul;
} tsLong;
```

Definition `blockdef tsLong tblkTimer1Sec _Timer1Sec;`

Example

```
decode on (_Timer1Sec)
{
    PollMyHardware ();
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 16 of: 187
Project: GSEOS		

4.2 Block `_Serial`

Description The system sends in the block `_Serial` the data received on the serial ports (COMx:).

Declaration

```
typedef struct
{
    ULONG ulPort;        // serial port number
    ULONG ulLength;     // length of block
    UCHAR auc[1024];    // block data
} tsSerial;
```

Definition `blockdef tsSerial tblkSerial _Serial;`

Example

```
decode on (_Serial)
{
    if (_Serial.ulPort == MYPORT)
        ProcessMyData (_Serial.ulLength, _Serial.auc);
}
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 17 of: 187
Project: GSEOS		

4.3 Block `_Net`

Description The system sends in the block `_Net` the data received on all network connections.

Declaration

```
typedef struct
{
    ULONG ulConnection;           // network connection number
    motorola union
    {
        ULONG ulPeerIP;          // Network packet source IP address
        UCHAR aucPeerIP[4];      // (peer address)
    };
    ULONG ulPort;                 // IP port, UDP source port
    ULONG ulDestinationPort;     // UDP destination port
    ULONG ulLength;              // length of received data
    UCHAR auc[1024*128];         // received block data
} tsNet;
```

Definition `blockdef tsNet tblkNet _Net;`

Example

```
decode on (_Net)
{
    switch (_Net.ulConnection)
    {
        case NET_BTIF_CLIENT:
        {
            ProcessBtifNetData (_Net.auc[0.._Net.ulLength]);
        }
        break;
    }
} // decode on (_Net)
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 18 of: 187
Project: GSEOS		

4.4 Block `_NetNotification`

Description The system informs about network events in the block `_NetNotification`.

Declaration

```
typedef struct
{
    ULONG ulConnection;    // network connection number
    ULONG ulNotification;  // NET_CONNECTED or NET_DISCONNECTED
    motorola union
    {
        ULONG ulPeerIP;    // Network packet source IP address
        UCHAR aucPeerIP[4]; // (peer address)
    };
    ULONG ulPort;          // IP port
} tsNetNotification;
```

Definition `blockdef tsNetNotification tblkNetNotification _NetNotification;`

Example

```
decode on (_NetNotification)
{
    switch (_NetNotification.ulConnection)
    {
        // test server
        case NET_BTIF_SERVER:
            switch (_NetNotification.ulNotification)
            {
                case NET_CONNECTED:
                    _Beep ();
                    break;

                case NET_DISCONNECTED:
                    // switch server back in listen mode for further
                    // connections
                    _NetOpen (NET_BTIF_SERVER);
                    break;
            }
    }
} // decode on (_NetNotification)
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 19 of: 187
Project: GSEOS		

4.5 Block `_Message`

Description The system sends all messages in the block `_Message`.

Declaration

```
typedef struct
{
    tsSystemTime sSystemTime;           // time of the message
    UCHAR        szCategory[16];       // message category
    UCHAR        szSource[16];         // message source
    UCHAR        szMessage[1024];     // message text
} tsMessage;
```

Definition `blockdef tsMessage tblkMessage _Message;`

Example `// you may record the message block for documentation`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 20 of: 187
Project: GSEOS		

5 Min/Max Functions

The min/max functions provided by GSEOS.g are for compatibility only. Please use the G-Language min/max operator <? and >?. For further information see “GSEOS Language Description”, IDA-GSEOS-0003.

```

ULONG minUL (ULONG u1, ULONG u2);
ULONG maxUL (ULONG u1, ULONG u2);
LONG minL (LONG l1, LONG l2);
LONG maxL (LONG l1, LONG l2);
USHORT minUS (USHORT us1, USHORT us2);
USHORT maxUS (USHORT us1, USHORT us2);
SHORT minS (SHORT s1, SHORT s2);
SHORT maxS (SHORT s1, SHORT s2);
UCHAR minUC (UCHAR uc1, UCHAR uc2);
UCHAR maxUC (UCHAR uc1, UCHAR uc2);
CHAR minC (CHAR c1, CHAR c2);
CHAR maxC (CHAR c1, CHAR c2);
DOUBLE minD (DOUBLE d1, DOUBLE d2);
DOUBLE maxD (DOUBLE d1, DOUBLE d2);

```

6 Command Channel Functions

The GSEOS uses command channels for all commanding operations. The command channels allow an easy change of the command destination. The function call is the same for all destinations. The only difference is the command channel identifier. The command channel identifier is an **unsigned long** to describe the command destination.

Table 3 Command Channels

Destination	Value	Channel Identifier	Description	
Internal	0x00ii0000	CMDCHANNEL_INTERNAL	ii	tbd
Hardware	0x01hhcc00	CMDCHANNEL_HARDWARE	hh	H/W identifier, depends on the project specific H/W driver
			cc	Command identifier, depends on the project specific H/W driver.
Serial	0x02pp0000	CMDCHANNEL_SERIAL	pp	The serial port (COMx:) number.
Network	0x03cc0000	CMDCHANNEL_NETWORK	cc	The network connection number.
File	0x04nn0000	CMDCHANNEL_FILE	nn	The file number.

The function **MakeCmdChannel ()** may be used to merge the channel identifier and sub identifier to the **unsigned long** of the command channel identifier.

```

ULONG MakeCmdChannel (UCHAR ucChannelId, UCHAR ucSub, UCHAR ucSubSub,
                     UCHAR ucSubSubSub)
{
    union
    {
        {
            ULONG ul;
            UCHAR uc[4];
        } u;

        u.uc[3] = ucChannelId;
        u.uc[2] = ucSub;
        u.uc[1] = ucSubSub;
        u.uc[0] = ucSubSubSub;

        return u.ul;
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 22 of: 187
Project: GSEOS		

6.1 `_SendCommand ()`

Prototype	<pre> BOOL _SendCommand (ULONG ulCmdChannel, CONST VOID &pvParameter, VOID &pvReturnBuffer, ULONG ulTimeout, ULONG ulElementSize, ULONG &pulBytesReturned, BOOL bWriteToLogFile); </pre>	
Description	The function sends a command to a specified command channel and receives (optionally) return values.	
Parameters	<code>ulCmdChannel</code>	The command channel (see description at section beginning).
	<code>pvParameter</code>	Points to the parameter buffer.
	<code>pvReturnBuffer</code>	Points to the buffer to receive return values.
	<code>ulTimeOut</code>	Timeout value in ms. The function waits up to the specified time for a successful return of the command channel function. On a timeout value of zero the function returns immediately.
	<code>ulElementSize</code>	Size of one element of the input parameter. Valid sizes are 1 (char), 2 (short), and 4 (long). The size is used for logging purposes only.
	<code>pulBytesReturned</code>	Number of bytes filled in <code>pvReturnBuffer</code> .
	<code>bWriteToLogFile</code>	TRUE Write the command, the parameter, and the return value in the message file. FALSE Don't log the command.
Return Value	TRUE	If the function succeeds, the return value is nonzero.
	FALSE	If the function fails, the return value is zero.
Example	<pre> BOOL ScsimRegisterRead (ULONG ulHwId, ULONG ulRegister, ULONG &pulRead) { ULONG aulParameter[2]; ULONG ulBytesReturned; aulParameter[0] = ulHwId; aulParameter[1] = ulRegister; return _SendCommand (MakeCmdChannel (CMDCHANNEL_HARDWARE, 0x04, 0x01, 0x00), aulParameter, pulRead, 0, sizeof (ULONG), ulBytesReturned, FALSE); } // ScsimRegisterRead </pre>	

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 23 of: 187
Project: GSEOS		

7 Dialog Input Functions

This section describes the various dialog boxes. The boxes are used to prompt the user for input. Some boxes accept a single value only (e.g. `_DlgGetLong ()`). In other boxes the user may input a space-separated list of values (e.g. `_DlgGetLongArray ()`).

All dialog boxes have a Cancel button. The function returns FALSE if either the ESC key is pressed or the Cancel button was selected.

7.1 _DlgGetChar ()

Prototype extern BOOL _DlgGetChar (CONST UCHAR &pszText[],
CHAR &c);

Description The function opens a dialog box for input of one byte (**char**) value.

Parameters pszText Points to a null-terminated string containing a description text to be displayed.

c Points to the return buffer (a single **char**).

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```
VOID TestDlgGetChar ()
{
    CHAR c;

    if (_DlgGetChar ("Enter Test Parameter.", c))
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.c = c;
        send (TestNew);
    }
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 25 of: 187
Project: GSEOS		

7.2 _DlgGetCharArray ()

Prototype extern BOOL _DlgGetCharArray (CONST UCHAR &pszText[],
 CHAR &pacBuffer[],
 ULONG ulElementCount,
 ULONG &pulValidElements);

Description The function opens a dialog box for input of an array of byte (**char**) values. The values must be separated by spaces.

Parameters

pszText	Points to a null-terminated string containing a description text to be displayed.
pacBuffer	Points to the return buffer (a char array).
ulElementCount	The length of the return buffer (in elements).
pulValidElements	Returns the count of received elements.

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```

VOID TestDlgGetCharArray ()
{
    CHAR  acBuffer[sizeof (tblkTest.ac)];
    ULONG ulElements, ulReturn, i;

    ulElements = sizeof (acBuffer);
    if (_DlgGetCharArray ("Enter Test Parameter.", acBuffer,
        ulElements, ulReturn))
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.ac = acBuffer;
        for (i = ulReturn; i < ulElements; ++i) // fill slack space
            TestNew.ac[i] = 0; // with zero
        send (TestNew);
    }
}

```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 27 of: 187
Project: GSEOS		

7.4 _DlgGetDoubleArray ()

Prototype	extern BOOL _DlgGetDoubleArray (CONST UCHAR &pszText[], DOUBLE &padBuffer[], ULONG ulElementCount, ULONG &pulValidElements);								
Description	The function opens a dialog box for input of an array of double values. The values must be separated by spaces.								
Parameters	<table> <tr> <td>pszText</td> <td>Points to a null-terminated string containing a description text to be displayed.</td> </tr> <tr> <td>padBuffer</td> <td>Points to the return buffer (a double array).</td> </tr> <tr> <td>ulElementCount</td> <td>The length of the return buffer (in elements).</td> </tr> <tr> <td>pulValidElements</td> <td>Returns the count of received elements.</td> </tr> </table>	pszText	Points to a null-terminated string containing a description text to be displayed.	padBuffer	Points to the return buffer (a double array).	ulElementCount	The length of the return buffer (in elements).	pulValidElements	Returns the count of received elements.
pszText	Points to a null-terminated string containing a description text to be displayed.								
padBuffer	Points to the return buffer (a double array).								
ulElementCount	The length of the return buffer (in elements).								
pulValidElements	Returns the count of received elements.								
Return Value	<table> <tr> <td>TRUE</td> <td>If the function succeeds, the return value is nonzero.</td> </tr> <tr> <td>FALSE</td> <td>If either the ESC key is pressed or the Cancel button was selected, the return value is zero.</td> </tr> </table>	TRUE	If the function succeeds, the return value is nonzero.	FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.				
TRUE	If the function succeeds, the return value is nonzero.								
FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.								
Example	<pre>VOID TestDlgGetDoubleArray () { DOUBLE adBuffer[sizeof (tblkTest.ad) / sizeof (tblkTest.ad[0])]; ULONG ulElements, ulReturn, i; ulElements = sizeof (adBuffer) / sizeof (adBuffer[0]); if (_DlgGetDoubleArray ("Enter Test Parameter.", adBuffer, ulElements, ulReturn)) { tblkTest TestNew; TestNew = Test; TestNew.ad = adBuffer; for (i = ulReturn; i < ulElements; ++i) // fill slack space TestNew.ad[i] = 0; // with zero send (TestNew); } }</pre>								

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 28 of: 187
Project: GSEOS		

7.5 _DlgGeti64 ()

Prototype extern BOOL _DlgGeti64 (CONST UCHAR &pszText[],
 tsINT64 &i64);

Description The function opens a dialog box for input of one 64-bit integer value.

Parameters pszText Points to a null-terminated string containing a description text to be displayed.

 l Points to the return buffer (a 64-bit integer).

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```

VOID TestDlgGeti64 ()
{
    tsINT64 i64;

    if (_DlgGeti64 ("Enter Test Parameter.", i64))
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.i64 = i64;
        send (TestNew);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 29 of: 187
Project: GSEOS		

7.6 _DlgGeti64Array ()

Prototype	<pre>extern BOOL _DlgGeti64Array (CONST UCHAR &pszText[], tsINT64 &pai64Buffer[], ULONG ulElementCount, ULONG &pulValidElements);</pre>	
Description	The function opens a dialog box for input of an array of 64-bit integer values. The values must be separated by spaces.	
Parameters	pszText	Points to a null-terminated string containing a description text to be displayed.
	palBuffer	Points to the return buffer (a 64-bit integer array).
	ulElementCount	The length of the return buffer (in elements).
	pulValidElements	Returns the count of received elements.
Return Value	TRUE	If the function succeeds, the return value is nonzero.
	FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.
Example	<pre>tblkTest TestNew = Test; static ULONG ulReturn = 1; if (_DlgGeti64Array ("Enter Test Parameter.", TestNew.ai64, sizeof (TestNew.ai64) / sizeof (TestNew.ai64[0]), ulReturn)) send (TestNew);</pre>	

IDA Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 30 of: 187

7.7 _DlgGetLong ()

Prototype extern BOOL _DlgGetLong (CONST UCHAR &pszText[],
LONG &l);

Description The function opens a dialog box for input of one **long** value.

Parameters pszText Points to a null-terminated string containing a description text to be displayed.

l Points to the return buffer (a single **long**).

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```
VOID TestDlgGetLong ()
{
    LONG l;

    if (_DlgGetLong ("Enter Test Parameter.", l))
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.l = l;
        send (TestNew);
    }
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 31 of: 187
Project: GSEOS		

7.8 _DlgGetLongArray ()

Prototype extern BOOL _DlgGetLongArray (CONST UCHAR &pszText[],
 LONG &palBuffer[],
 ULONG ulElementCount,
 ULONG &pulValidElements);

Description The function opens a dialog box for input of an array of **long** values. The values must be separated by spaces.

Parameters

pszText	Points to a null-terminated string containing a description text to be displayed.
palBuffer	Points to the return buffer (a long array).
ulElementCount	The length of the return buffer (in elements).
pulValidElements	Returns the count of received elements.

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```

VOID TestDlgGetLongArray ()
{
    LONG  alBuffer[sizeof (tblkTest.al) / sizeof (tblkTest.al[0])];
    ULONG ulElements, ulReturn, i;

    ulElements = sizeof (alBuffer) / sizeof (alBuffer[0]);
    if (_DlgGetLongArray ("Enter Test Parameter.", alBuffer,
                          ulElements, ulReturn))
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.al = alBuffer;
        for (i = ulReturn; i < ulElements; ++i) // fill slack space
            TestNew.al[i] = 0;                // with zero
        send (TestNew);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 32 of: 187
Project: GSEOS		

7.9 _DlgGetOpenFileName ()

Prototype	extern BOOL _DlgGetOpenFileName (UCHAR &pszBuffer[], ULONG ulBufferLen);	
Description	The function creates an “Open” common dialog box that lets the user specify a file name.	
Parameters	pszBuffer	Points to the return buffer for the file name. The file name is null-terminated.
	ulBufferLen	Maximum length of the return buffer.
Return Value	TRUE	If the function succeeds, the return value is nonzero.
	FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.
Example	<pre>VOID TestDlgGetOpenFileName () { UCHAR aucBuffer[1024]; if (_DlgGetOpenFileName (aucBuffer, sizeof (aucBuffer))) _MessageBox ("Filename:", aucBuffer, MB_OK MB_ICONINFORMATION); }</pre>	

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 33 of: 187
Project: GSEOS		

7.10 _DlgGetSaveFileName ()

Prototype	extern BOOL _DlgGetSaveFileName (UCHAR &pszBuffer[], ULONG ulBufferLen);	
Description	The function creates an “Save As” common dialog box that lets the user specify a file name.	
Parameters	pszBuffer	Points to the return buffer for the file name. The file name is null-terminated.
	ulBufferLen	Maximum length of the return buffer.
Return Value	TRUE	If the function succeeds, the return value is nonzero.
	FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.
Example	<pre>VOID TestDlgGetSaveFileName () { UCHAR aucBuffer[1024]; if (_DlgGetSaveFileName (aucBuffer, sizeof (aucBuffer))) _MessageBox ("Filename:", aucBuffer, MB_OK MB_ICONINFORMATION); }</pre>	

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1 style="margin: 0;">GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 34 of: 187
Project: GSEOS		

7.11 _DlgGetShort ()

Prototype extern BOOL _DlgGetShort (CONST UCHAR &pszText[],
 SHORT &s);

Description The function opens a dialog box for input of one **short** value.

Parameters pszText Points to a null-terminated string containing a description text to be displayed.

 s Points to the return buffer (a single **short**).

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```

VOID TestDlgGetShort ()
{
    SHORT s;

    if (_DlgGetShort ("Enter Test Parameter.", s))
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.s = s;
        send (TestNew);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 35 of: 187
Project: GSEOS		

7.12 _DlgGetShortArray ()

Prototype	extern BOOL _DlgGetShortArray (CONST UCHAR &pszText[], SHORT &pasBuffer[], ULONG ulElementCount, ULONG &pulValidElements);								
Description	The function opens a dialog box for input of an array of short values. The values must be separated by spaces.								
Parameters	<table> <tr> <td>pszText</td> <td>Points to a null-terminated string containing a description text to be displayed.</td> </tr> <tr> <td>pasBuffer</td> <td>Points to the return buffer (a short array).</td> </tr> <tr> <td>ulElementCount</td> <td>The length of the return buffer (in elements).</td> </tr> <tr> <td>pulValidElements</td> <td>Returns the count of received elements.</td> </tr> </table>	pszText	Points to a null-terminated string containing a description text to be displayed.	pasBuffer	Points to the return buffer (a short array).	ulElementCount	The length of the return buffer (in elements).	pulValidElements	Returns the count of received elements.
pszText	Points to a null-terminated string containing a description text to be displayed.								
pasBuffer	Points to the return buffer (a short array).								
ulElementCount	The length of the return buffer (in elements).								
pulValidElements	Returns the count of received elements.								
Return Value	<table> <tr> <td>TRUE</td> <td>If the function succeeds, the return value is nonzero.</td> </tr> <tr> <td>FALSE</td> <td>If either the ESC key is pressed or the Cancel button was selected, the return value is zero.</td> </tr> </table>	TRUE	If the function succeeds, the return value is nonzero.	FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.				
TRUE	If the function succeeds, the return value is nonzero.								
FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.								
Example	<pre>VOID TestDlgGetShortArray () { SHORT asBuffer[sizeof (tblkTest.as) / sizeof (tblkTest.as[0])]; ULONG ulElements, ulReturn, i; ulElements = sizeof (asBuffer) / sizeof (asBuffer[0]); if (_DlgGetShortArray ("Enter Test Parameter.", asBuffer, ulElements, ulReturn)) { tblkTest TestNew; TestNew = Test; TestNew.as = asBuffer; for (i = ulReturn; i < ulElements; ++i) // fill slack space TestNew.as[i] = 0; // with zero send (TestNew); } }</pre>								

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 36 of: 187
Project: GSEOS		

7.13 _DlgGetString ()

Prototype extern BOOL _DlgGetString (CONST UCHAR &pszText[],
 UCHAR &pszBuffer[],
 ULONG ulElementCount,
 ULONG &pulValidElements);

Description The function opens a dialog box for input of a string.

Parameters

pszText	Points to a null-terminated string containing a description text to be displayed.
pszBuffer	Points to the return buffer for the null-terminated string.
ulElementCount	The length of the return buffer (in elements).
pulValidElements	Returns the count of received elements.

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```

VOID TestDlgGetString ()
{
    UCHAR aucBuffer[512];
    ULONG ulReturn;

    if (_DlgGetString ("Enter Test Parameter.", aucBuffer,
                      sizeof (aucBuffer), ulReturn))
    {
        _MessageBox ("String:", aucBuffer,
                    MB_OK | MB_ICONINFORMATION);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 37 of: 187
Project: GSEOS		

7.14 _DlgGetUChar ()

- Prototype** extern BOOL _DlgGetUChar (CONST UCHAR &pszText[],
 UCHAR &uc);
- Description** The function opens a dialog box for input of one unsigned byte (**unsigned char**) value.
- Parameters** pszText Points to a null-terminated string containing a description text to be displayed.

 uc Points to the return buffer (a single **unsigned char**).
- Return Value** TRUE If the function succeeds, the return value is nonzero.

 FALSE If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```

VOID TestDlgGetUChar ()
{
    UCHAR uc;

    if (_DlgGetUChar ("Enter Test Parameter.", uc))
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.uc = uc;
        send (TestNew);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 38 of: 187
Project: GSEOS		

7.15 _DlgGetUCharArray ()

Prototype	extern BOOL _DlgGetUCharArray (CONST UCHAR &pszText[], UCHAR &paucBuffer[], ULONG ulElementCount, ULONG &pulValidElements);								
Description	The function opens a dialog box for input of an array of unsigned byte (unsigned char) values. The values must be separated by spaces.								
Parameters	<table> <tr> <td>pszText</td> <td>Points to a null-terminated string containing a description text to be displayed.</td> </tr> <tr> <td>paucBuffer</td> <td>Points to the return buffer (an unsigned char array).</td> </tr> <tr> <td>ulElementCount</td> <td>The length of the return buffer (in elements).</td> </tr> <tr> <td>pulValidElements</td> <td>Returns the count of received elements.</td> </tr> </table>	pszText	Points to a null-terminated string containing a description text to be displayed.	paucBuffer	Points to the return buffer (an unsigned char array).	ulElementCount	The length of the return buffer (in elements).	pulValidElements	Returns the count of received elements.
pszText	Points to a null-terminated string containing a description text to be displayed.								
paucBuffer	Points to the return buffer (an unsigned char array).								
ulElementCount	The length of the return buffer (in elements).								
pulValidElements	Returns the count of received elements.								
Return Value	<table> <tr> <td>TRUE</td> <td>If the function succeeds, the return value is nonzero.</td> </tr> <tr> <td>FALSE</td> <td>If either the ESC key is pressed or the Cancel button was selected, the return value is zero.</td> </tr> </table>	TRUE	If the function succeeds, the return value is nonzero.	FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.				
TRUE	If the function succeeds, the return value is nonzero.								
FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.								
Example	<pre>VOID TestDlgGetUCharArray () { CHAR aucBuffer[sizeof (tblkTest.auc)]; ULONG ulElements, ulReturn, i; ulElements = sizeof (aucBuffer); if (_DlgGetUCharArray ("Enter Test Parameter.", aucBuffer, ulElements, ulReturn)) { tblkTest TestNew; TestNew = Test; TestNew.auc = aucBuffer; for (i = ulReturn; i < ulElements; ++i) // fill slack space TestNew.auc[i] = 0; // with zero send (TestNew); } }</pre>								

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 39 of: 187
Project: GSEOS		

7.16 _DlgGetui64 ()

Prototype extern BOOL _DlgGetui64 (CONST UCHAR &pszText[],
 tsUINT64 &ui64);

Description The function opens a dialog box for input of one unsigned 64-bit integer value.

Parameters pszText Points to a null-terminated string containing a description text to be displayed.

 l Points to the return buffer (an unsigned 64-bit integer).

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```

VOID TestDlgGetui64 ()
{
    tsUINT64 ui64;

    if (_DlgGetui64 ("Enter Test Parameter.", ui64))
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.ui64 = ui64;
        send (TestNew);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 40 of: 187
Project: GSEOS		

7.17 _DlgGetui64Array ()

Prototype	<pre>extern BOOL _DlgGetui64Array (CONST UCHAR &pszText[], tsUINT64 &pui64Buffer[], ULONG ulElementCount, ULONG &pulValidElements);</pre>	
Description	The function opens a dialog box for input of an array of unsigned 64-bit integer values. The values must be separated by spaces.	
Parameters	pszText	Points to a null-terminated string containing a description text to be displayed.
	palBuffer	Points to the return buffer (an unsigned 64-bit integer array).
	ulElementCount	The length of the return buffer (in elements).
	pulValidElements	Returns the count of received elements.
Return Value	TRUE	If the function succeeds, the return value is nonzero.
	FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.
Example	<pre>tblkTest TestNew = Test; static ULONG ulReturn = 1; if (_DlgGetui64Array ("Enter Test Parameter.", TestNew.aui64, sizeof (TestNew.aui64) / sizeof (TestNew.aui64[0]), ulReturn)) send (TestNew);</pre>	

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 41 of: 187
Project: GSEOS		

7.18 _DlgGetULong ()

Prototype extern BOOL _DlgGetULong (CONST UCHAR &pszText[],
 ULONG &ul);

Description The function opens a dialog box for input of one **unsigned long** value.

Parameters pszText Points to a null-terminated string containing a description text to be displayed.

 ul Points to the return buffer (a single **unsigned long**).

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```

VOID TestDlgGetULong ()
{
    ULONG ul;

    if (_DlgGetULong ("Enter Test Parameter.", ul))
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.ul = ul;
        send (TestNew);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 42 of: 187
Project: GSEOS		

7.19 _DlgGetULongArray ()

Prototype	extern BOOL _DlgGetULongArray (CONST UCHAR &pszText[], ULONG &paulBuffer[], ULONG ulElementCount, ULONG &pulValidElements);								
Description	The function opens a dialog box for input of an array of unsigned long values. The values must be separated by spaces.								
Parameters	<table> <tr> <td style="padding-right: 20px;">pszText</td> <td>Points to a null-terminated string containing a description text to be displayed.</td> </tr> <tr> <td>paulBuffer</td> <td>Points to the return buffer (an unsigned long array).</td> </tr> <tr> <td>ulElementCount</td> <td>The length of the return buffer (in elements).</td> </tr> <tr> <td>pulValidElements</td> <td>Returns the count of received elements.</td> </tr> </table>	pszText	Points to a null-terminated string containing a description text to be displayed.	paulBuffer	Points to the return buffer (an unsigned long array).	ulElementCount	The length of the return buffer (in elements).	pulValidElements	Returns the count of received elements.
pszText	Points to a null-terminated string containing a description text to be displayed.								
paulBuffer	Points to the return buffer (an unsigned long array).								
ulElementCount	The length of the return buffer (in elements).								
pulValidElements	Returns the count of received elements.								
Return Value	<table> <tr> <td style="padding-right: 20px;">TRUE</td> <td>If the function succeeds, the return value is nonzero.</td> </tr> <tr> <td>FALSE</td> <td>If either the ESC key is pressed or the Cancel button was selected, the return value is zero.</td> </tr> </table>	TRUE	If the function succeeds, the return value is nonzero.	FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.				
TRUE	If the function succeeds, the return value is nonzero.								
FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.								
Example	<pre>VOID TestDlgGetULongArray () { ULONG aulBuffer[sizeof (tblkTest.aul) / sizeof (tblkTest.aul[0])]; ULONG ulElements, ulReturn, i; ulElements = sizeof (aulBuffer) / sizeof (aulBuffer[0]); if (_DlgGetULongArray ("Enter Test Parameter.", aulBuffer, ulElements, ulReturn)) { tblkTest TestNew; TestNew = Test; TestNew.aul = aulBuffer; for (i = ulReturn; i < ulElements; ++i) // fill slack space TestNew.aul[i] = 0; // with zero send (TestNew); } }</pre>								

IDA Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 43 of: 187

7.20 _DlgGetUShort ()

Prototype extern BOOL _DlgGetUShort (CONST UCHAR &pszText[],
USHORT &us);

Description The function opens a dialog box for input of one **unsigned short** value.

Parameters pszText Points to a null-terminated string containing a description text to be displayed.

us Points to the return buffer (a single **unsigned short**).

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If either the ESC key is pressed or the Cancel button was selected, the return value is zero.

Example

```
VOID TestDlgGetUShort ()  
{  
    USHORT us;  
  
    if (_DlgGetUShort ("Enter Test Parameter.", us))  
    {  
        tblkTest TestNew;  
  
        TestNew = Test;  
        TestNew.us = us;  
        send (TestNew);  
    }  
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 44 of: 187
Project: GSEOS		

7.21 _DlgGetUShortArray ()

Prototype	extern BOOL _DlgGetUShortArray (CONST UCHAR &pszText[], USHORT &pausBuffer[], ULONG ulElementCount, ULONG &pulValidElements);								
Description	The function opens a dialog box for input of an array of unsigned short values. The values must be separated by spaces.								
Parameters	<table> <tr> <td style="padding-right: 20px;">pszText</td> <td>Points to a null-terminated string containing a description text to be displayed.</td> </tr> <tr> <td>pausBuffer</td> <td>Points to the return buffer (an unsigned short array).</td> </tr> <tr> <td>ulElementCount</td> <td>The length of the return buffer (in elements).</td> </tr> <tr> <td>pulValidElements</td> <td>Returns the count of received elements.</td> </tr> </table>	pszText	Points to a null-terminated string containing a description text to be displayed.	pausBuffer	Points to the return buffer (an unsigned short array).	ulElementCount	The length of the return buffer (in elements).	pulValidElements	Returns the count of received elements.
pszText	Points to a null-terminated string containing a description text to be displayed.								
pausBuffer	Points to the return buffer (an unsigned short array).								
ulElementCount	The length of the return buffer (in elements).								
pulValidElements	Returns the count of received elements.								
Return Value	<table> <tr> <td style="padding-right: 20px;">TRUE</td> <td>If the function succeeds, the return value is nonzero.</td> </tr> <tr> <td>FALSE</td> <td>If either the ESC key is pressed or the Cancel button was selected, the return value is zero.</td> </tr> </table>	TRUE	If the function succeeds, the return value is nonzero.	FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.				
TRUE	If the function succeeds, the return value is nonzero.								
FALSE	If either the ESC key is pressed or the Cancel button was selected, the return value is zero.								
Example	<pre> VOID TestDlgGetUShortArray () { USHORT ausBuffer[sizeof (tblkTest.aus) / sizeof (tblkTest.aus[0])]; ULONG ulElements, ulReturn, i; ulElements = sizeof (ausBuffer) / sizeof (ausBuffer[0]); if (_DlgGetUShortArray ("Enter Test Parameter.", ausBuffer, ulElements, ulReturn)) { tblkTest TestNew; TestNew = Test; TestNew.aus = ausBuffer; for (i = ulReturn; i < ulElements; ++i) // fill slack space TestNew.aus[i] = 0; // with zero send (TestNew); } } </pre>								

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1 style="margin: 0;">GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 45 of: 187
Project: GSEOS		

8 Network Functions

The GSEOS is able to handle up to 256 network connections at the same time. Every network connection has a separate configuration. The GSEOS supports TCP/IP and UDP protocol. Other protocols may be implemented in the future on demand.

The configuration is set by the function `_NetConfig ()`. The function needs a structure of type `_tsNetConfigStruct` filled with the configuration:

```
typedef struct
{
    ULONG    ulSize;                // not longer used, compatibility only
    union
    {
        BOOL    bIsServer;         // compatibility only:
        // FALSE: client (connect to other server);
        // TRUE: listen (wait for clients)
        NetProtocol eNetProtocol;  // NET_TCP_CLIENT, connect to other server
        // NET_TCP_SERVER, listen(wait for clients)
        // NET_UDP, UDP
    };
    UCHAR    szHostName[MAX_PATH]; // host name or address
    USHORT   usPort;               // IP port, UDP source port
    USHORT   usDestinationPort;   // UDP destination port
    BOOL     bSendKeepAlive;       // send keep-alive packets
} _tsNetConfigStruct;
```

The GSEOS stores the network configuration. It is possible to close the network configuration and open it again without a new `_NetConfig ()` call. Note: Port numbers below 1024 are reserved (ftp, telnet, http, etc.).

After setting the configuration the network connection is ready to open with the function `_NetOpen ()`. On a connection configured as TCP client, the GSEOS makes a connection to the server. On a connection configured as TCP server, the GSEOS goes to a listen state and awaits incoming connections at the IP specified port.

If the function `_NetOpen ()` was called inside a batch file, the network connection will be closed automatically on batch end or batch abort.

The open (connected) network connection is ready to send and receive data. The command channel `CMDCHANNEL_NETWORK` sends the data to the network connection. For an easy access there is a function `NetSend ()` defined in `gseos.g` (see section 8.5).

The system sends the data received on all network connections in the block `_Net`. For the description of this block see section 4.3 Block `_Net`.

Network events like “connect” or “disconnect” are sent the Block `_NetNotification` described in section 4.4 Block `_NetNotification`.

The network connection may be closed (disconnected) by `_NetClose ()`. It is not a failure to close the connection more than once. In this case the function does nothing. The GSEOS closes all still open network connections on program exit.

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 46 of: 187
Project: GSEOS		

8.1 `_NetClose ()`

Prototype	<code>extern BOOL _NetClose (UCHAR ucConnection, BOOL bGracefulDisconnect);</code>	
Description	The function closes (disconnects) a network connection. It is not a failure to close an already closed connection.	
Parameters	<code>ucConnection</code>	The network connection number.
	<code>bGracefulDisconnect</code>	TRUE: Close graceful (shutdown socket, read all pending data, close socket). The function may block some time on bad connections! FALSE: Makes a simple disconnect. The function never blocks with if this parameter is set to FALSE.
Return Value	<code>TRUE</code>	If the function succeeds, the return value is nonzero.
	<code>FALSE</code>	If the function failed, the return value is zero.
Example	<code>_NetClose (NET_BTIF_SERVER, FALSE);</code>	

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 49 of: 187
Project: GSEOS		

8.3 `_NetIsOpen ()`

Prototype `extern BOOL _NetIsOpen (UCHAR ucConnection);`

Description The function determines the network connection state.

Parameters `ucConnection` The network connection number.

Return Value `TRUE` The network connection is connected (open).

`FALSE` The network connection is disconnected (closed), or listening.

Example `if (_NetIsOpen (NET_BTIF_SERVER))
 NetSend (NET_BTIF_SERVER, "test", TRUE);`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 50 of: 187
Project: GSEOS		

8.4 `_NetOpen ()`

Prototype extern BOOL `_NetOpen (UCHAR ucConnection);`

Description The function opens (connects) a network connection. The configuration of the network connection must be set with the function `_NetConfig ()`. It is a failure to open an already open connection.

On a connection configured as TCP client, the GSEOS makes a connection to the server. On a connection configured as TCP server, the GSEOS goes to a listen state and awaits incoming connections at the IP port. On a connection configured as UDP, the GSEOS opens the UDP port.

If the function was called from a batch file, the connection will be closed automatically on batch end or batch abort.

Parameters ucConnection The network connection number.

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If the function failed, the return value is zero.

Example

```


VOID NetTestConfig ()
{
    _tsNetConfigStruct sNetConfigStruct;

    // test client
    sNetConfigStruct.ulSize           = sizeof (sNetConfigStruct);
    sNetConfigStruct.bIsServer       = TRUE;
    sNetConfigStruct.szHostName      = ""; // the server is this
                                        // local machine, so it
                                        // needs no name
    sNetConfigStruct.usPort          = 1024;
    sNetConfigStruct.bSendKeepAlive  = FALSE;
    _NetConfig (NET_BTIF_SERVER, sNetConfigStruct);

    // test server
    sNetConfigStruct.ulSize           = sizeof (sNetConfigStruct);
    sNetConfigStruct.bIsServer       = FALSE;
    sNetConfigStruct.szHostName      = "134.169.29.162"; // host to
                                        // connect (the name of
                                        // the server at the
                                        // other side)
    sNetConfigStruct.usPort          = 1024;
    sNetConfigStruct.bSendKeepAlive  = FALSE;
    _NetConfig (NET_BTIF_CLIENT, sNetConfigStruct);

    _NetOpen (NET_BTIF_SERVER); // server in listen mode
} // NetTestConfig

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 51 of: 187
Project: GSEOS		

8.5 NetSend ()

Prototype	<pre> BOOL NetSend (UCHAR ucConnection, CONST VOID &pvParameter, BOOL bWriteToLogFile); </pre>										
Description	<p>The function sends data to an open network connection. The data will be sent to the network command channel, therefore the function is defined in G-Language and not as real Runtime Library function.</p>										
Parameters	<table> <tr> <td style="padding-right: 20px;">ucConnection</td> <td>The network connection.</td> </tr> <tr> <td>pvParameter</td> <td>Points to the values to send.</td> </tr> <tr> <td>bWriteToLogFile</td> <td> <table> <tr> <td style="padding-right: 20px;">TRUE</td> <td>Write the command, the parameter, and the return value in the message file.</td> </tr> <tr> <td>FALSE</td> <td>Don't log the command.</td> </tr> </table> </td> </tr> </table>	ucConnection	The network connection.	pvParameter	Points to the values to send.	bWriteToLogFile	<table> <tr> <td style="padding-right: 20px;">TRUE</td> <td>Write the command, the parameter, and the return value in the message file.</td> </tr> <tr> <td>FALSE</td> <td>Don't log the command.</td> </tr> </table>	TRUE	Write the command, the parameter, and the return value in the message file.	FALSE	Don't log the command.
ucConnection	The network connection.										
pvParameter	Points to the values to send.										
bWriteToLogFile	<table> <tr> <td style="padding-right: 20px;">TRUE</td> <td>Write the command, the parameter, and the return value in the message file.</td> </tr> <tr> <td>FALSE</td> <td>Don't log the command.</td> </tr> </table>	TRUE	Write the command, the parameter, and the return value in the message file.	FALSE	Don't log the command.						
TRUE	Write the command, the parameter, and the return value in the message file.										
FALSE	Don't log the command.										
Return Value	<table> <tr> <td style="padding-right: 20px;">TRUE</td> <td>If the function succeeds, the return value is nonzero.</td> </tr> <tr> <td>FALSE</td> <td>If the function failed, the return value is zero.</td> </tr> </table>	TRUE	If the function succeeds, the return value is nonzero.	FALSE	If the function failed, the return value is zero.						
TRUE	If the function succeeds, the return value is nonzero.										
FALSE	If the function failed, the return value is zero.										
Example	<pre> decode on (_BTIF) { static UCHAR Sync[] = "BTIF"; static ULONG ulCount; tsNetBtif NetBtif; ++ulCount; NetBtif.aucSync = Sync[0..3]; NetBtif.ulCount = ulCount; NetBtif.BTIF = _BTIF; NetSend (NET_BTIF_SERVER, NetBtif.auc, FALSE); } // decode on (_BTIF) </pre>										

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 52 of: 187
Project: GSEOS		

9 Serial Port Functions

The GSEOS is able to handle up to 256 serial ports (COM0 - COM255) at the same time. Every port has a separate configuration.

The configuration is set by the function `_SerialConfig ()`. The function needs a structure of type `_tsSerialConfigStruct` filled with the configuration:

```
typedef struct
{
    ULONG ulSize;                // sizeof (_tsSerialConfigStruct)
    ULONG ulBaudRate;           // 110, 300, 1200, ...
    ULONG ulDataBits;           // [4..8]
    ULONG ulParity;              // 0-4 = none, odd, even, mark, space
    ULONG ulStopBits;           // 0-2 = 1, 1.5, 2
    ULONG ulFlowControl;        // 0-2 = none, Xon/Xoff, H/W
    ULONG ulTxBuffer;           // default: 4096 bytes
    ULONG ulRxBuffer;           // default: 4096 bytes
    ULONG ulTxTimeout;          // default: 1000 ms
    ULONG ulRxTimeout;          // default: 1000 ms
    ULONG ulRxGapDetectionTime; // default: 500 ms
    BOOL bInitialDTR;           // default: TRUE (set)
    BOOL bInitialRTS;           // default: FALSE (clear)
} _tsSerialConfigStruct;
```

The GSEOS stores the port configuration. It is possible to close the port and open it again without a new `_SerialConfig ()` call.

After setting the configuration the port is ready to open with the function `_SerialOpen ()`. If another application already use the port the GSEOS can't open it.

If the function `_SerialOpen ()` was called inside a batch file, the port will be closed automatically on batch end or batch abort.

The open port is ready to send and receive data. The command channel `CMDCHANNEL_SERIAL` sends the data to the port. For an easy access there is a function `SerialSend ()` defined in `gseos.g` (see section 9.6).

The system sends the data received on the serial ports in the block `_Serial`. For the description of this block see section 4.2 Block `_Serial`.

The port may be closed by `_SerialClose ()`. It is not a failure to close the port more than once. In this case the function does nothing. The GSEOS closes all still open ports on program exit.

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 53 of: 187
Project: GSEOS		

9.1 _SerialClose ()

Prototype extern BOOL _SerialClose (UCHAR ucPort);

Description The function closes a serial port. It is not a failure to close an already closed port.

Parameters ucPort The serial port number.

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If the function failed, the return value is zero.

Example

```

VOID DlgSerialSendText (UCHAR ucPort)
{
    UCHAR aucBuffer[1024];
    ULONG ulReturn;
    BOOL bOpen;

    if (_DlgGetString ("Enter string to send to Serial Port.",
                      aucBuffer, sizeof (aucBuffer) - 2,
                      ulReturn))
    {
        // append CR LF
        aucBuffer[ulReturn] = '\r';
        aucBuffer[ulReturn+1] = '\n';

        // open, send, restore previous port state
        bOpen = _SerialIsOpen (ucPort);
        _SerialOpen (ucPort);
        SerialSend (ucPort, FALSE, aucBuffer[0, ulReturn + 2], TRUE);
        if (!bOpen)
            _SerialClose (ucPort);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 54 of: 187
Project: GSEOS		

9.2 `_SerialConfig ()`

Prototype extern BOOL `_SerialConfig (UCHAR ucPort,
 CONST _tsSerialConfigStruct &sSerialConfigStruct);`

Description The function stores the serial port configuration. The configuration is used on the next `_SerialOpen ()` call. The port must be in closed state to set a new configuration.

Parameters

<code>ucPort</code>	The serial port number.
<code>sSerialConfigStruct</code>	The configuration of the port. See description in section 9 Serial Port Functions.

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	If the function failed, the return value is zero.

Example

```
void _main ()
{
    _tsSerialConfigStruct sSerialConfigStruct;

    // init OSIRIS remote power supply com port
    sSerialConfigStruct.ulSize      = sizeof (sSerialConfigStruct);
    sSerialConfigStruct.ulBaudRate  = 9600;
    sSerialConfigStruct.ulDataBits  = 8; // [4..8]
    sSerialConfigStruct.ulParity    = 0; // 0-4=none,o,e,mark,space
    sSerialConfigStruct.ulStopBits  = 0; // 0-2=1,1.5,2
    sSerialConfigStruct.ulFlowControl= 0; // 0-2=none,Xon/Xoff,H/W
    sSerialConfigStruct.ulTxBuffer  = 4096; // bytes
    sSerialConfigStruct.ulRxBuffer  = 4096; // bytes
    sSerialConfigStruct.ulTxTimeout = 1000; // ms
    sSerialConfigStruct.ulRxTimeout = 1000; // ms
    sSerialConfigStruct.ulRxGapDetectionTime = 500; // ms
    sSerialConfigStruct.bInitialDTR = TRUE;
    sSerialConfigStruct.bInitialRTS = FALSE;

    _SerialConfig (PORT_POWERSUPPLY, sSerialConfigStruct);
    _SerialOpen (PORT_POWERSUPPLY);
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 55 of: 187
Project: GSEOS		

9.3 _SerialIsOpen ()

Prototype extern BOOL _SerialIsOpen (UCHAR ucPort);

Description The function determines the serial port state.

Parameters ucPort The serial port number.

Return Value TRUE The port is open.

 FALSE The port is closed.

Example

```

VOID DlgSerialSendText (UCHAR ucPort)
{
    UCHAR aucBuffer[1024];
    ULONG ulReturn;
    BOOL bOpen;

    if (_DlgGetString ("Enter string to send to Serial Port.",
                      aucBuffer, sizeof (aucBuffer) - 2,
                      ulReturn))
    {
        // append CR LF
        aucBuffer[ulReturn] = '\r';
        aucBuffer[ulReturn+1] = '\n';

        // open, send, restore previous port state
        bOpen = _SerialIsOpen (ucPort);
        _SerialOpen (ucPort);
        SerialSend (ucPort, FALSE, aucBuffer, ulReturn + 2, TRUE);
        if (!bOpen)
            _SerialClose (ucPort);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 56 of: 187
Project: GSEOS		

9.4 `_SerialEscapeFunction ()`

Prototype	<code>extern BOOL _SerialEscapeFunction (UCHAR ucPort, ULONG ulFunction);</code>	
Description	The function directs a specified communications device to perform an extended function.	
Parameters	<code>ucPort</code>	The serial port number.
	<code>ulFunction</code>	Specifies the code of the extended function to perform. This parameter can be one of the following values:
	<code>SETXOFF</code>	Simulate XOFF received.
	<code>SETXON</code>	Simulate XON received.
	<code>SETRTS</code>	Set RTS high.
	<code>CLRRTS</code>	Set RTS low.
	<code>SETDTR</code>	Set DTR high.
	<code>CLRDTR</code>	Set DTR low.
	<code>RESETDEV</code>	Reset device if possible.
	<code>SETBREAK</code>	Set the device break line.
	<code>CLRBREAK</code>	Clear the device break line.
Return Value	<code>TRUE</code>	If the function succeeds, the return value is nonzero.
	<code>FALSE</code>	If the function failed, the return value is zero.
Example	<pre>while (1) { _SerialEscapeFunction (2, SETDTR); sleep (1000); _SerialEscapeFunction (2, CLRDTR); sleep (1000); }</pre>	

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 57 of: 187
Project: GSEOS		

9.5 _SerialOpen ()

Prototype extern BOOL _SerialOpen (UCHAR ucPort);

Description The function opens a serial port. The configuration of the port must be set with the function **_SerialConfig ()**. It is a failure to open an already open port. If the function was called inside a batch file, the port will be closed automatically on batch end or batch abort.

Parameters ucPort The serial port number.

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If the function failed, the return value is zero.

Example

```
void _main ()
{
    _tsSerialConfigStruct sSerialConfigStruct;

    // init OSIRIS remote power supply com port
    sSerialConfigStruct.ulSize      = sizeof (sSerialConfigStruct);
    sSerialConfigStruct.ulBaudRate  = 9600;
    sSerialConfigStruct.ulDataBits  = 8; // [4..8]
    sSerialConfigStruct.ulParity     = 0; // 0-4=none,o,e,mark,space
    sSerialConfigStruct.ulStopBits  = 0; // 0-2=1,1.5,2
    sSerialConfigStruct.ulFlowControl= 0; // 0-2=none,Xon/Xoff,H/W
    sSerialConfigStruct.ulTxBuffer   = 4096; // bytes
    sSerialConfigStruct.ulRxBuffer   = 4096; // bytes
    sSerialConfigStruct.ulTxTimeout  = 1000; // ms
    sSerialConfigStruct.ulRxTimeout  = 1000; // ms
    sSerialConfigStruct.ulRxGapDetectionTime = 500; // ms
    sSerialConfigStruct.bInitialDTR  = TRUE;
    sSerialConfigStruct.bInitialRTS  = FALSE;

    _SerialConfig (PORT_POWERSUPPLY, sSerialConfigStruct);
    _SerialOpen (PORT_POWERSUPPLY);
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 58 of: 187
Project: GSEOS		

9.6 SerialSend ()

Prototype	<pre> BOOL SerialSend (UCHAR ucPort, CONST VOID &pvParameter, BOOL bWriteToLogFile); </pre>	
Description	The function sends data to an open serial port. The data will be sent to the serial command channel, therefore the function is defined in G-Language and not as real Runtime Library function.	
Parameters	ucPort	The serial port number.
	pvParameter	Points to the values to send.
	bWriteToLogFile	TRUE Write the command, the parameter, and the return value in the message file. FALSE Don't log the command.
Return Value	TRUE	If the function succeeds, the return value is nonzero.
	FALSE	If the function failed, the return value is zero.
Example	<pre> SerialSend (PORT_POWER SUPPLY, "*CLS;\r\n", TRUE); </pre>	

10 File Functions

The GSEOS is able to handle up to 256 open files at the same time. Every file has a separate configuration.

The configuration is set by the function **_FileConfig ()**. The function needs a structure of type **_tsFileConfigStruct** filled with the configuration:

```
typedef struct
{
    ULONG ulSize;                // sizeof (_tsFileConfigStruct)
    UCHAR szName[MAX_PATH];     // name of the file
    UCHAR szMode[8];            // open mode (like C-function fopen())
} _tsFileConfigStruct;
```

The mode string specifies the type of access requested for the file, as follows:

Table 4 Mode String Values

Characters in mode string	Description		
	opens file for	if file exist	if file does not exist
r	read	opens file	function fails
w	write	set file length to zero	creates an empty file
r+	read and write	opens file	function fails
w+	read and write	set file length to zero	creates an empty file
+, rw, rw+	read and write	opens file	creates an empty file
R	Share file for read operations. Subsequent open operations on the file will succeed only if read access is requested.		
W	Share file for write operations. Subsequent open operations on the file will succeed only if write access is requested		
c	Commit write operations directly to disk. Instructs the system to write through any intermediate cache and go directly to disk. The system can still cache write operations, but cannot lazily flush them.		
a	Not longer supported. Use _FileSeek() function to move file pointer to the end of the file.		
b	Not longer supported. File is opened always in binary mode (i.e. without carriage-return, linefeed, and CTRL+Z character translation).		
t	Not longer supported. File is opened always in binary mode (i.e. without carriage-return, linefeed, and CTRL+Z character translation).		

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1 style="margin: 0;">GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 60 of: 187
Project: GSEOS		

Unknown characters in the mode string will be ignored. The order of the characters in the mode string doesn't matter (e.g. "rw+R" is equal to "+wRr").

The GSEOS stores the file configuration. It is possible to close the file and open it again without a new **_FileConfig ()** call.

After setting the configuration the file is ready to open with the function **_FileOpen ()**. If another application already use the file the GSEOS maybe can't open it. It depends on the sharing flags used by the GSEOS and the other application.

If the function **_FileOpen ()** was called inside a batch file, the file will be closed automatically on batch end or batch abort.

The open file is ready to read and write data. The command channel **CMDCHANNEL_FILE** sends the data to the file. For an easy access there is a function **FileWrite ()** defined in **gseos.g** (see section 10.15).

The file must be closed by **_FileClose ()**. The function clears the buffers for the specified file and causes all buffered data to be written to the file. It is not a failure to close the file more than once. In this case the function does nothing. The GSEOS closes all still open files on program exit.

There are several functions for file reading (**_FileReadXxx ()**). Some of them are able to read from file, make packets, and translate the values given in ASCII format into a binary array (**_FileReadAscIntoXxxArray ()**). The file has to be in the following format:

```
//
// 1355 Test File
//
// Syntax: - comment:      "///", the comment can start in any position,
//                    and extends until the next new line
//          - prefixes:    "0x" or "0X" is hex
//                    "0b" or "0B" is bin
//                    leading "0" is oct
//                    no prefix is decimal
//                    or double format (e.g. -1.3e-10)
//          - packet end:  '*', NOTE: You don't need the last '*'
//                    at the end of file.
//          - empty lines and any count of spaces and tabs are allowed

// packet 1
0x10000000 0x20000000 // normal packet
0x33333333 0x12345678
*

// packet 2
* // an empty packet

// packet 3 and 4
1 2 * 3 4 * // two packets in one line

// packet 5
0x1DA
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 61 of: 187
Project: GSEOS		

10.1 _FileClose ()

Prototype extern BOOL _FileClose (UCHAR ucNumber);

Description The function closes a file, clears the buffers for the specified file and causes all buffered data to be written to the file. It is not a failure to close an already closed file.

Parameters ucNumber The file number.

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If the function failed, the return value is zero.

Example

```

VOID TestDlgWriteFile ()
{
    UCHAR aucBuffer[1024];
    ULONG ulReturn;

    if (_DlgGetString ("Enter Text.", aucBuffer, sizeof (aucBuffer),
                      ulReturn))
    {
        _tsFileConfigStruct sFileConfigStruct;

        sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
        sFileConfigStruct.szName = "\\data\\test.txt"; // default path
                                                // is the GSEOS.EXE path
        sFileConfigStruct.szMode = "w+";
        _FileConfig (FILE_TEST, sFileConfigStruct);
        _FileOpen (FILE_TEST);
        FileWrite (FILE_TEST, aucBuffer[0], ulReturn, TRUE);
        _FileClose (FILE_TEST);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 62 of: 187
Project: GSEOS		

10.2 _FileConfig ()

Prototype	extern BOOL _FileConfig (UCHAR ucNumber, CONST _tsFileConfigStruct &sFileConfigStruct);	
Description	The function stores the file configuration. The configuration is used on the next _FileOpen () call. The file must be in closed state to set a new configuration.	
Parameters	ucNumber	The file number.
	sFileConfigStruct	The configuration of the file. See description in section 10 File Functions.
Return Value	TRUE	If the function succeeds, the return value is nonzero.
	FALSE	If the function failed, the return value is zero.
Example	<pre> VOID TestDlgWriteFile () { UCHAR aucBuffer[1024]; ULONG ulReturn; if (_DlgGetString ("Enter Text.", aucBuffer, sizeof (aucBuffer), ulReturn)) { _tsFileConfigStruct sFileConfigStruct; sFileConfigStruct.ulSize = sizeof (sFileConfigStruct); sFileConfigStruct.szName = "\\data\\test.txt"; // default path // is the GSEOS.EXE path sFileConfigStruct.szMode = "w+"; _FileConfig (FILE_TEST, sFileConfigStruct); _FileOpen (FILE_TEST); FileWrite (FILE_TEST, aucBuffer[0], ulReturn, TRUE); _FileClose (FILE_TEST); } } </pre>	

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 63 of: 187
Project: GSEOS		

10.3 _FileIsOpen ()

Prototype extern BOOL _FileIsOpen (UCHAR ucNumber);

Description The function determines the file state.

Parameters ucNumber The file number.

Return Value TRUE The file is open.

 FALSE The file is closed.

Example

```

VOID Test_FileIsOpen ()
{
    if (_FileIsOpen (FILE_TEST))
    {
        FileWrite (FILE_TEST, "End", TRUE);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 64 of: 187
Project: GSEOS		

10.4 _FileOpen ()

Prototype extern BOOL _FileOpen (UCHAR ucNumber);

Description The function opens a file. The configuration of the file must be set with the function **_FileConfig ()**. It is a failure to open an already open file. If the function was called inside a batch file, the file will be closed automatically on batch end or batch abort.

Parameters ucNumber The file number.

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If the function failed, the return value is zero.

Example

```

VOID TestDlgWriteFile ()
{
    UCHAR aucBuffer[1024];
    ULONG ulReturn;

    if (_DlgGetString ("Enter Text.", aucBuffer, sizeof (aucBuffer),
                      ulReturn))
    {
        _tsFileConfigStruct sFileConfigStruct;

        sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
        sFileConfigStruct.szName = "\\data\\test.txt"; // default path
                                                // is the GSEOS.EXE path
        sFileConfigStruct.szMode = "w+";
        _FileConfig (FILE_TEST, sFileConfigStruct);
        _FileOpen (FILE_TEST);
        FileWrite (FILE_TEST, aucBuffer[0], ulReturn, TRUE);
        _FileClose (FILE_TEST);
    }
}

```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 65 of: 187
Project: GSEOS		

10.5 _FileRead ()

Prototype extern BOOL _FileRead (UCHAR ucNumber,
UCHAR &paucBuffer[],
ULONG ulElementCount,
ULONG &pulValidElements);

Description The function read an array of **unsigned char** values from file.

Parameters

ucNumber	The file number.
paucBuffer	Points to the return buffer (an unsigned char array).
ulElementCount	The length of the return buffer (in elements). This is the count to read.
pulValidElements	Returns the count of read elements.

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function failed, the return value is zero.

Example

```
VOID TestReadFile ()
{
    UCHAR aucBuffer[sizeof (tblkTest.aucFileString)];
    ULONG ulReturn, i;
    _tsFileConfigStruct sFileConfigStruct;

    sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
    sFileConfigStruct.szName = "\\data\\test.txt"; // default path
                                         // is the GSEOS.EXE path

    sFileConfigStruct.szMode = "r";
    _FileConfig (FILE_TEST, sFileConfigStruct);
    _FileOpen (FILE_TEST);
    _FileRead (FILE_TEST, aucBuffer, sizeof (aucBuffer), ulReturn);
    _FileClose (FILE_TEST);
    if (ulReturn > 0)
    {
        tblkTest TestNew;

        TestNew = Test;
        TestNew.aucFileString = aucBuffer;
        TestNew.aucFileString[ulReturn] = '\\0'; // make zero
                                                // terminated string

        send (TestNew);
    }
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004
		Issue: 1.12 Date: 5/25/2004
Project: GSEOS		Page: 66 of: 187

10.6 _FileReadAscIntoCharArray ()

Prototype

```
extern BOOL _FileReadAscIntoCharArray (UCHAR ucNumber,
                                       CHAR &pacBuffer[],
                                       ULONG ulElementCount,
                                       ULONG &pulValidElements);
```

Description The function read an ASCII file, makes packets, and translates the values given in ASCII format into a binary array of byte (**char**) values. The file has to be in a format described in section 10 File Functions.

Parameters

- ucNumber The file number.
- pacBuffer Points to the return buffer (a **char** array).
- ulElementCount The length of the return buffer (in elements). This is the count to read.
- pulValidElements Returns the count of read elements.

Return Value

- TRUE If the function succeeds, the return value is nonzero.
- FALSE End of file (EOF) reached or other error occurred.

Example

```
void DlgFesimPMCmdSendHkFileAsc (ULONG ulHwId)
{
    ULONG ulReturn;
    UCHAR aucFileName[MAX_PATH];
    _tsFileConfigStruct sFileConfigStruct;
    union {
        UCHAR auc[1024];
        struct {
            ULONG ulHwId;
            CHAR acBuffer[0];
        };
    } sParameter;

    if (_DlgGetOpenFileName (aucFileName, sizeof (aucFileName)))
    {
        sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
        sFileConfigStruct.szName = aucFileName;
        sFileConfigStruct.szMode = "r";
        _FileConfig (FILE_FESIMPMHK, sFileConfigStruct);
        _FileOpen (FILE_FESIMPMHK);

        while (_FileReadAscIntoCharArray (FILE_FESIMPMHK,
                                          sParameter.acBuffer, sizeof (sParameter.ac) -
                                          sizeof (ULONG), ulReturn))
        {
            sParameter.ulHwId = ulHwId; // send packet to H/W
            FesimPMCmdSendHk (sParameter.auc, ulReturn +
                             sizeof (unsigned long), 1);
        }
        _FileClose (FILE_FESIMPMHK);
    }
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 67 of: 187
Project: GSEOS		

10.7 _FileReadAscIntoDoubleArray ()

Prototype extern BOOL _FileReadAscIntoDoubleArray (UCHAR ucNumber,
DOUBLE &padBuffer[],
ULONG ulElementCount,
ULONG &pulValidElements);

Description The function read an ASCII file, makes packets, and translates the values given in ASCII format into a binary array of **double** values. The file has to be in a format described in section 10 File Functions.

Parameters

ucNumber	The file number.
padBuffer	Points to the return buffer (a double array).
ulElementCount	The length of the return buffer (in elements). This is the count to read.
pulValidElements	Returns the count of read elements.

Return Value TRUE If the function succeeds, the return value is nonzero.
FALSE End of file (EOF) reached or other error occurred.

Example

```

_tsFileConfigStruct sFileConfigStruct;
DOUBLE              ad[10];
ULONG              ulReturn, i;
UCHAR              szBuffer[128];

sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
sFileConfigStruct.szName = "\\data\\test.txt";
sFileConfigStruct.szMode = "r";
_FileConfig (FILE_TEST, sFileConfigStruct);
_FileOpen (FILE_TEST);

while (_FileReadAscIntoDoubleArray (FILE_TEST, ad, 10, ulReturn))
{
    for (i = 0; i < ulReturn; ++i)
    {
        _dtoa (ad[i], szBuffer);
        _MessageBox ("Test", szBuffer, MB_OK);
    }
    _MessageBox ("Test", "Packet end.", MB_OK);
}

// file:
// 1.2 3.4 *
// 5.6e-10
// output:
// 1.2
// 3.4
// "Packet end."
// 5.6e-10
// "Packet end."

```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 69 of: 187
Project: GSEOS		

10.9 _FileReadAscIntoLongArray ()

Prototype extern BOOL _FileReadAscIntoLongArray (UCHAR ucNumber,
LONG &palBuffer[],
ULONG ulElementCount,
ULONG &pulValidElements);

Description The function read an ASCII file, makes packets, and translates the values given in ASCII format into a binary array of **long** values. The file has to be in a format described in section 10 File Functions.

Parameters

ucNumber	The file number.
palBuffer	Points to the return buffer (a long array).
ulElementCount	The length of the return buffer (in elements). This is the count to read.
pulValidElements	Returns the count of read elements.

Return Value TRUE If the function succeeds, the return value is nonzero.
FALSE End of file (EOF) reached or other error occurred.

Example

```
void DlgScsiml355WriteAscFile ()
{
    UCHAR aucFileName[MAX_PATH];
    ULONG ulReturn;
    _tsFileConfigStruct sFileConfigStruct;
    union {
        ULONG aul[1024];
        struct {
            ULONG ulEop;
            LONG alBuffer[0];
        };
    } sParameter;

    if (_DlgGetOpenFileName (aucFileName, sizeof (aucFileName)))
    {
        sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
        sFileConfigStruct.szName = aucFileName;
        sFileConfigStruct.szMode = "r";
        _FileConfig (FILE_SCSIM1355ASC, sFileConfigStruct);
        _FileOpen (FILE_SCSIM1355ASC);

        while (_FileReadAscIntoLongArray (FILE_SCSIM1355ASC,
            Parameter.alBuffer, sizeof (sParameter.aul) /
            sizeof (sParameter.aul[0]) - 1, ulReturn))
        {
            sParameter.ulEop = 1; // send EOP1 packet
            Scsiml355Write (sParameter.aul, ulReturn + 1);
        }
        _FileClose (FILE_SCSIM1355ASC);
    }
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 70 of: 187
Project: GSEOS		

10.10 `_FileReadAscIntoShortArray ()`

Prototype `extern BOOL _FileReadAscIntoShortArray (UCHAR ucNumber,
SHORT &pasBuffer[],
ULONG ulElementCount,
ULONG &pulValidElements);`

Description The function read an ASCII file, makes packets, and translates the values given in ASCII format into a binary array of **short** values. The file has to be in a format described in section 10 File Functions.

Parameters

<code>ucNumber</code>	The file number.
<code>pasBuffer</code>	Points to the return buffer (a short array).
<code>ulElementCount</code>	The length of the return buffer (in elements). This is the count to read.
<code>pulValidElements</code>	Returns the count of read elements.

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	End of file (EOF) reached or other error occurred.

Example

```
// see _FileReadAscIntoByteArray ()
// or _FileReadAscIntoLongArray ()
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 71 of: 187
Project: GSEOS		

10.11 _FileReadAscIntoUCharArray ()

Prototype extern BOOL _FileReadAscIntoUCharArray (UCHAR ucNumber,
 UCHAR &paucBuffer[],
 ULONG ulElementCount,
 ULONG &pulValidElements);

Description The function read an ASCII file, makes packets, and translates the values given in ASCII format into a binary array of unsigned byte (**unsigned char**) values. The file has to be in a format described in section 10 File Functions.

Parameters

ucNumber	The file number.
paucBuffer	Points to the return buffer (an unsigned char array).
ulElementCount	The length of the return buffer (in elements). This is the count to read.
pulValidElements	Returns the count of read elements.

Return Value TRUE If the function succeeds, the return value is nonzero.
 FALSE End of file (EOF) reached or other error occurred.

Example

```

void DlgFesimPMCmdSendHkFileAsc (ULONG ulHwId)
{
    ULONG ulReturn;
    UCHAR aucFileName[MAX_PATH];
    _tsFileConfigStruct sFileConfigStruct;
    union {
        UCHAR auc[1024];
        struct {
            ULONG ulHwId;
            UCHAR aucBuffer[0];
        };
    } sParameter;

    if (_DlgGetOpenFileName (aucFileName, sizeof (aucFileName)))
    {
        sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
        sFileConfigStruct.szName = aucFileName;
        sFileConfigStruct.szMode = "r";
        _FileConfig (FILE_FESIMPMHK, sFileConfigStruct);
        _FileOpen (FILE_FESIMPMHK);

        while (_FileReadAscIntoUCharArray (FILE_FESIMPMHK,
            sParameter.aucBuffer, sizeof (sParameter.auc) -
            sizeof (ULONG), ulReturn))
        {
            sParameter.ulHwId = ulHwId; // send packet to H/W
            FesimPMCmdSendHk (sParameter.auc, ulReturn +
                sizeof (unsigned long), 1);
        }
        _FileClose (FILE_FESIMPMHK);
    }
}

```

10.12 `_FileReadAscIntoui64Array ()`

Prototype

```
extern BOOL _FileReadAscIntoui64Array (UCHAR ucNumber,
                                       tsUINT64 &paui64Buffer[],
                                       ULONG ulElementCount,
                                       ULONG &pulValidElements);
```

Description The function read an ASCII file, makes packets, and translates the values given in ASCII format into a binary array of unsigned 64-bit integer values. The file has to be in a format described in section 10 File Functions.

Parameters

<code>ucNumber</code>	The file number.
<code>paui64Buffer</code>	Points to the return buffer (an unsigned 64-bit integer array).
<code>ulElementCount</code>	The length of the return buffer (in elements). This is the count to read.
<code>pulValidElements</code>	Returns the count of read elements.

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	End of file (EOF) reached or other error occurred.

Example

```
_tsFileConfigStruct sFileConfigStruct;
tsUINT64 aui64[8];
ULONG ulReturn, i;
UCHAR szBuffer0[128], szBuffer1[128];

sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
sFileConfigStruct.szName = "\\data\\test.txt";
sFileConfigStruct.szMode = "rt";
_FileConfig (FILE_TEST, sFileConfigStruct);
_FileOpen (FILE_TEST);

while (_FileReadAscIntoui64Array (FILE_TEST, aui64, 8, ulReturn))
{
    _ultoa (ulReturn, szBuffer1, 10);
    _strcat (szBuffer1, " values read.");
    _AddMessage (MSG_INFORMATION, "Test", szBuffer1);

    _strcpy (szBuffer1, "file contains: ");
    for (i = 0; i < ulReturn; ++i)
    {
        _ui64toa (aui64[i], szBuffer0, 10);
        _strcat (szBuffer1, szBuffer0);
        _strcat (szBuffer1, " ");
    }

    _AddMessage (MSG_INFORMATION, "Test", szBuffer1);
}

FileClose (FILE_TEST);}
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 73 of: 187
Project: GSEOS		

10.13 `_FileReadAscIntoULongArray ()`

Prototype extern BOOL `_FileReadAscIntoULongArray` (UCHAR ucNumber,
 ULONG &paulBuffer[],
 ULONG ulElementCount,
 ULONG &pulValidElements);

Description The function read an ASCII file, makes packets, and translates the values given in ASCII format into a binary array of **unsigned long** values. The file has to be in a format described in section 10 File Functions.

Parameters

<code>ucNumber</code>	The file number.
<code>paulBuffer</code>	Points to the return buffer (an unsigned long array).
<code>ulElementCount</code>	The length of the return buffer (in elements). This is the count to read.
<code>pulValidElements</code>	Returns the count of read elements.

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	End of file (EOF) reached or other error occurred.

Example

```
void DlgScsiml355WriteAscFile ()
{
    UCHAR aucFileName[MAX_PATH];
    ULONG ulReturn;
    _tsFileConfigStruct sFileConfigStruct;
    union {
        ULONG aul[1024];
        struct {
            ULONG ulEop;
            ULONG aulBuffer[0];
        };
    } sParameter;

    if (_DlgGetOpenFileName (aucFileName, sizeof (aucFileName)))
    {
        sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
        sFileConfigStruct.szName = aucFileName;
        sFileConfigStruct.szMode = "r";
        _FileConfig (FILE_SCSIM1355ASC, sFileConfigStruct);
        _FileOpen (FILE_SCSIM1355ASC);

        while (_FileReadAscIntoLongArray (FILE_SCSIM1355ASC,
            Parameter.aulBuffer, sizeof (sParameter.aul) /
            sizeof (sParameter.aul[0]) - 1, ulReturn))
        {
            sParameter.ulEop = 1; // send EOP1 packet
            Scsiml355Write (sParameter.aul, ulReturn + 1);
        }
        _FileClose (FILE_SCSIM1355ASC);
    }
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 74 of: 187
Project: GSEOS		

10.14 `_FileReadAscIntoUShortArray ()`

Prototype extern BOOL `_FileReadAscIntoUShortArray` (UCHAR ucNumber,
 USHORT &pausBuffer[],
 ULONG ulElementCount,
 ULONG &pulValidElements);

Description The function read an ASCII file, makes packets, and translates the values given in ASCII format into a binary array of **unsigned short** values. The file has to be in a format described in section 10 File Functions.

Parameters

<code>ucNumber</code>	The file number.
<code>pausBuffer</code>	Points to the return buffer (an unsigned short array).
<code>ulElementCount</code>	The length of the return buffer (in elements). This is the count to read.
<code>pulValidElements</code>	Returns the count of read elements.

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	End of file (EOF) reached or other error occurred.

Example // see `_FileReadAscIntoUByteArray ()`
 // or `_FileReadAscIntoULongArray ()`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 75 of: 187
Project: GSEOS		

10.15 `_FileSeek ()`

Prototype `BOOL _FileSeek (UCHAR ucNumber,
 CONST tsINT64 &i64Offset,
 ULONG ulMoveMethod);`

Description The function sets the file pointer to a new position that is `i64Offset` bytes from the file location given by `ulMoveMethod`.

Parameters

<code>ucNumber</code>	The file number.
<code>i64Offset</code>	IN: file pointer offset to set as 64-bit integer OUT: current file pointer as 64-bit integer
<code>ulMoveMethod</code>	<code>FILE_BEGIN</code> Beginning of file. <code>FILE_CURRENT</code> Current position of file pointer. <code>FILE_END</code> End of file.

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	If the function failed, the return value is zero.

Example

```

_tsFileConfigStruct sFileConfigStruct;
tsINT64 i64;

sFileConfigStruct.ulSize = sizeof (sFileConfigStruct);
sFileConfigStruct.szName = "\\data\\test.txt";
sFileConfigStruct.szMode = "w+";
_FileConfig (FILE_TEST, sFileConfigStruct);
_FileOpen (FILE_TEST);

FileWrite (FILE_TEST, "123"[0,3], TRUE);

// move 1 byte back from file end
i64.ulLow = -1;
i64.lHigh = -1;
_FileSeek (FILE_TEST, i64, FILE_END);

FileWrite (FILE_TEST, "abc"[0,3], TRUE);

_FileClose (FILE_TEST);

// now in file: "12abc"

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 76 of: 187
Project: GSEOS		

10.16 FileWrite ()

Prototype `BOOL FileWrite (UCHAR ucNumber,
 CONST VOID &pvParameter,
 BOOL bWriteToLogFile);`

Description The function writes data to an open file. The data will be sent to the file command channel, therefore the function is defined in G-Language and not as real Runtime Library function.

Parameters

<code>ucNumber</code>	The file number.				
<code>pvParameter</code>	Points to the values to write.				
<code>bWriteToLogFile</code>	<table> <tr> <td style="padding-right: 20px;">TRUE</td> <td>Write the command, the parameter, and the return value in the message file.</td> </tr> <tr> <td style="padding-right: 20px;">FALSE</td> <td>Don't log the command.</td> </tr> </table>	TRUE	Write the command, the parameter, and the return value in the message file.	FALSE	Don't log the command.
TRUE	Write the command, the parameter, and the return value in the message file.				
FALSE	Don't log the command.				

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function failed, the return value is zero.

Example `FileWrite (FILE_TEST, "Test\r\n", TRUE);`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 77 of: 187

11 Converting Functions

This section describes the various converting functions.

The functions `_AscToXxxArray ()` are able to convert values given as a null-terminated string in ASCII format into a binary array. The values in the string has to be separated by spaces or commas (“,”).

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 78 of: 187
Project: GSEOS		

11.1 _AscToCharArray ()

Prototype extern BOOL _AscToCharArray (CONST UCHAR &pszString[],
CHAR &pacBuffer[],
ULONG ulElementCount,
ULONG &pulValidElements);

Description The function converts values given as a null-terminated string in ASCII format into a byte (**char**) array. The values in the string has to be separated by spaces or commas (“,”).

Parameters

pszString	The string to convert.
pacBuffer	Points to the return buffer (a char array).
ulElementCount	The length of the return buffer (in elements).
pulValidElements	Returns the count of converted elements.

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function fails, the return value is zero.

Example

```
void ScsimSciWriteCmdStr (UCHAR &pszParameterText[])
{
    union
    {
        {
            UCHAR auc[1024];
            struct
            {
                UCHAR ucLength[2];    // length in words (motorola)
                CHAR  acBuffer[0];
            };
        } sParameter;
    } union
    {
        {
            ULONG ul;
            UCHAR uc[4];
        } sLength;
    }
    ULONG ulReturn;

    if (_AscToCharArray (pszParameterText, sParameter.acBuffer,
        sizeof (sParameter) - 2, ulReturn))
    {
        // set length in words (motorola)
        sLength.ul = (ulReturn + 1) / 2;
        sParameter.ucLength[0] = sLength.uc[1];
        sParameter.ucLength[1] = sLength.uc[0];
        ScsimSciWriteCmd (sParameter.auc, ulReturn + 2);
    }
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 79 of: 187
Project: GSEOS		

11.2 `_AscToDoubleArray ()`

Prototype	<pre>extern BOOL _AscToDoubleArray (CONST UCHAR &pszString[], DOUBLE &padBuffer[], ULONG ulElementCount, ULONG &pulValidElements);</pre>								
Description	<p>The function converts values given as a null-terminated string in ASCII format into a double array. The values in the string has to be separated by spaces or commas (“,”).</p>								
Parameters	<table> <tr> <td style="padding-right: 20px;"><code>pszString</code></td> <td>The string to convert.</td> </tr> <tr> <td><code>padBuffer</code></td> <td>Points to the return buffer (a double array).</td> </tr> <tr> <td><code>ulElementCount</code></td> <td>The length of the return buffer (in elements).</td> </tr> <tr> <td><code>pulValidElements</code></td> <td>Returns the count of converted elements.</td> </tr> </table>	<code>pszString</code>	The string to convert.	<code>padBuffer</code>	Points to the return buffer (a double array).	<code>ulElementCount</code>	The length of the return buffer (in elements).	<code>pulValidElements</code>	Returns the count of converted elements.
<code>pszString</code>	The string to convert.								
<code>padBuffer</code>	Points to the return buffer (a double array).								
<code>ulElementCount</code>	The length of the return buffer (in elements).								
<code>pulValidElements</code>	Returns the count of converted elements.								
Return Value	<table> <tr> <td style="padding-right: 20px;"><code>TRUE</code></td> <td>If the function succeeds, the return value is nonzero.</td> </tr> <tr> <td><code>FALSE</code></td> <td>If the function fails, the return value is zero.</td> </tr> </table>	<code>TRUE</code>	If the function succeeds, the return value is nonzero.	<code>FALSE</code>	If the function fails, the return value is zero.				
<code>TRUE</code>	If the function succeeds, the return value is nonzero.								
<code>FALSE</code>	If the function fails, the return value is zero.								
Example	<pre>DOUBLE adBuffer[5]; ULONG ulReturn, i; _AscToDoubleArray ("1.2e3 -42.0", adBuffer, sizeof (adBuffer) / sizeof (adBuffer[0]), ulReturn); for (i = 0; i < ulReturn; ++i) MyFunction (adBuffer[i]);</pre>								

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004
		Issue: 1.12 Date: 5/25/2004
Project: GSEOS		Page: 80 of: 187

11.3 _AscToLongArray ()

Prototype extern BOOL _AscToLongArray (CONST UCHAR &pszString[],
 LONG &palBuffer[],
 ULONG ulElementCount,
 ULONG &pulValidElements);

Description The function converts values given as a null-terminated string in ASCII format into a **long** array. The values in the string has to be separated by spaces or commas (“,”).

Parameters

pszString	The string to convert.
palBuffer	Points to the return buffer (a long array).
ulElementCount	The length of the return buffer (in elements).
pulValidElements	Returns the count of converted elements.

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function fails, the return value is zero.

Example

```
void Scsim1355WriteStr (UCHAR &pszParameterText[])
{
    union
    {
        ULONG aul[1024];
        struct
        {
            ULONG ulEop;
            LONG  alBuffer[0];
        };
    } sParameter;
    ULONG ulReturn;

    if (_AscToLongArray (pszParameterText, sParameter.alBuffer,
                        sizeof (sParameter.aul) /
                        sizeof (sParameter.aul[0]) - 1,
                        ulReturn))
    {
        sParameter.ulEop = 1; // EOP1
        Scsim1355Write (sParameter.aul, ulReturn + 1);
    }
}
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 81 of: 187
Project: GSEOS		

11.4 `_AscToShortArray ()`

Prototype

```
extern BOOL _AscToShortArray (CONST UCHAR &pszString[],
                              SHORT &pasBuffer[],
                              ULONG ulElementCount,
                              ULONG &pulValidElements);
```

Description The function converts values given as a null-terminated string in ASCII format into a **short** array. The values in the string has to be separated by spaces or commas (“,”).

Parameters

<code>pszString</code>	The string to convert.
<code>pasBuffer</code>	Points to the return buffer (a short array).
<code>ulElementCount</code>	The length of the return buffer (in elements).
<code>pulValidElements</code>	Returns the count of converted elements.

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	If the function fails, the return value is zero.

Example `// see void _AscToByteArray () or _AscToLongArray ()`

11.5 _AscToUCharArray ()

Prototype extern BOOL _AscToUCharArray (CONST UCHAR &pszString[],
UCHAR &paucBuffer[],
ULONG ulElementCount,
ULONG &pulValidElements);

Description The function converts values given as a null-terminated string in ASCII format into an unsigned byte (**unsigned char**) array. The values in the string has to be separated by spaces or commas (“,”).

Parameters

pszString	The string to convert.
paucBuffer	Points to the return buffer (an unsigned char array).
ulElementCount	The length of the return buffer (in elements).
pulValidElements	Returns the count of converted elements.

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function fails, the return value is zero.

Example

```
void ScsimSciWriteCmdStr (UCHAR &pszParameterText[])
{
    union
    {
        UCHAR auc[1024];
        struct
        {
            UCHAR ucLength[2];    // length in words (motorola)
            UCHAR aucBuffer[0];
        };
    } sParameter;
    union
    {
        ULONG ul;
        UCHAR uc[4];
    } sLength;
    ULONG ulReturn;

    if (_AscToUCharArray (pszParameterText, sParameter.aucBuffer,
        sizeof (sParameter) - 2, ulReturn))
    {
        // set length in words (motorola)
        sLength.ul = (ulReturn + 1) / 2;
        sParameter.ucLength[0] = sLength.uc[1];
        sParameter.ucLength[1] = sLength.uc[0];
        ScsimSciWriteCmd (sParameter.auc, ulReturn + 2);
    }
}
```

11.6 _AscToUlongArray ()

Prototype extern BOOL _AscToUlongArray (CONST UCHAR &pszString[],
 ULONG &paulBuffer[],
 ULONG ulElementCount,
 ULONG &pulValidElements);

Description The function converts values given as a null-terminated string in ASCII format into an **unsigned long** array. The values in the string has to be separated by spaces or commas (“,”).

Parameters

pszString	The string to convert.
paulBuffer	Points to the return buffer (an unsigned long array).
ulElementCount	The length of the return buffer (in elements).
pulValidElements	Returns the count of converted elements.

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function fails, the return value is zero.

Example

```
void Scsim1355WriteStr (UCHAR &pszParameterText[])
{
    union
    {
        ULONG aul[1024];
        struct
        {
            ULONG ulEop;
            ULONG aulBuffer[0];
        };
    } sParameter;
    ULONG ulReturn;

    if (_AscToUlongArray (pszParameterText, sParameter.aulBuffer,
        sizeof (sParameter.aul) /
        sizeof (sParameter.aul[0]) - 1,
        ulReturn))
    {
        sParameter.ulEop = 1; // EOP1
        Scsim1355Write (sParameter.aul, ulReturn + 1);
    }
}
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1 style="margin: 0;">GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 84 of: 187
Project: GSEOS		

11.7 _AscToUShortArray ()

Prototype `extern BOOL _AscToUShortArray (CONST UCHAR &pszString[],
 USHORT &pausBuffer[],
 ULONG ulElementCount,
 ULONG &pulValidElements);`

Description The function converts values given as a null-terminated string in ASCII format into an **unsigned short** array. The values in the string has to be separated by spaces or commas (“,”).


Parameters

- `pszString` The string to convert.
- `pausBuffer` Points to the return buffer (an **unsigned short** array).
- `ulElementCount` The length of the return buffer (in elements).
- `pulValidElements` Returns the count of converted elements.

Return Value

- `TRUE` If the function succeeds, the return value is nonzero.
- `FALSE` If the function fails, the return value is zero.

Example `// see void _AscToUByteArray () or _AscToUlongArray ()`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004
		Issue: 1.12 Date: 5/25/2004
Project: GSEOS		Page: 85 of: 187

11.8 _atod ()

Prototype extern BOOL _atod (CONST UCHAR &pszString[],
 DOUBLE &d);

Description Converts a null-terminated string pszString to a floating-point **double** number d. This function recognizes the character representation of a floating-point number in the following order:

- An optional string of tabs and spaces
- An optional sign
- A string of digits and an optional decimal point (the digits can be on both sides of the decimal point)
- An optional e or E followed by an optional signed integer
- An optional trailing string of tabs and spaces

The characters must match this generic format:

[whitespace] [sign] [ddd] [.] [ddd] [e|E[sign]ddd] [whitespace]

The function also recognizes +INF and -INF for plus and minus infinity, and +NAN and -NAN for Not-a-Number.

In this function, unrecognized character returns an error code.

Parameters pszString The string to convert.

 d Points to the return buffer (a **double**).

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If the function fails, the return value is zero.

Example

```
DOUBLE d;

_atod ("1.2e-10", d);

if (d != 1.2e-10)
    _Beep ();
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 86 of: 187
Project: GSEOS		

11.9 `_atoi64()`

Prototype `extern BOOL _atoi64 (CONST UCHAR &pszString[],
 tsINT64 &i64);`

Description Converts a null-terminated string `pszString` to a 64-bit integer number `i64`. This function recognizes the character representation of a integer number in the following order:

- An optional string of tabs and spaces
- An optional sign
- An optional prefix “0x” or “0X” for hex values
- An optional prefix “0b” or “0B” for binary values
- An optional prefix “0” for octal values
- A string of digits
- An optional trailing string of tabs and spaces

The characters must match this generic format:

`[whitespace] [[sign] | [“0x”|“0X”] | [“0b”|“0B”]] [ddd] [whitespace]`

In this function, unrecognized character returns an error code.

Parameters `pszString` The string to convert.

`i64` Points to the return buffer (a 64-bit integer).

Return Value `TRUE` If the function succeeds, the return value is nonzero.

`FALSE` If the function fails, the return value is zero.

Example

```
tsINT64 i64;

_atoi64 ("-42", i64);

if ((i64.ulLow != -42) || (i64.lHigh != -1))
    _Beep ();
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 87 of: 187

11.10 `_atol ()`

Prototype `extern BOOL _atol (CONST UCHAR &pszString[],
 LONG &l);`

Description Converts a null-terminated string `pszString` to a **long** number `l`. This function recognizes the character representation of a integer number in the following order:

- An optional string of tabs and spaces
- An optional sign
- An optional prefix “0x” or “0X” for hex values
- An optional prefix “0b” or “0B” for binary values
- An optional prefix “0” for octal values
- A string of digits
- An optional trailing string of tabs and spaces

The characters must match this generic format:

```
[whitespace] [[sign] | [{"0x"}|"0X"} | [{"0b"}|"0B"}]] [ddd] [whitespace]
```

In this function, unrecognized character returns an error code.

Parameters

<code>pszString</code>	The string to convert.
<code>l</code>	Points to the return buffer (a long).

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	If the function fails, the return value is zero.

Example

```
LONG l;  
  
_atol ("-42", l);  
  
if (l != -42)  
    _Beep ();
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 88 of: 187
Project: GSEOS		

11.11 `_atoui64 ()`

Prototype extern BOOL `_atoui64 (CONST UCHAR &pszString[],
 tsUINT64 &ui64);`

Description Converts a null-terminated string `pszString` to a 64-bit unsigned integer number `ui64`. This function recognizes the character representation of a integer number in the following order:

- An optional string of tabs and spaces
- An optional sign
- An optional prefix "0x" or "0X" for hex values
- An optional prefix "0b" or "0B" for binary values
- An optional prefix "0" for octal values
- A string of digits
- An optional trailing string of tabs and spaces

The characters must match this generic format:

`[whitespace] [[sign] | [{"0x"}|"0X"] | [{"0b"}|"0B"]]] [ddd] [whitespace]`

In this function, unrecognized character returns an error code.

Parameters

<code>pszString</code>	The string to convert.
<code>ui64</code>	Points to the return buffer (a 64-bit unsigned integer).

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function fails, the return value is zero.

Example

```
tsUINT64 ui64;

_atoui64 ("42", ui64);

if ((ui64.ulLow != 42) || (ui64.ulHigh != 0))
    _Beep ();
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004
		Issue: 1.12 Date: 5/25/2004
Project: GSEOS		Page: 89 of: 187

11.12 `_atoul ()`

Prototype `extern BOOL _atoul (CONST UCHAR &pszString[],
ULONG &ul);`

Description Converts a null-terminated string `pszString` to an **unsigned long** number `ul`. This function recognizes the character representation of a integer number in the following order:

- An optional string of tabs and spaces
- An optional sign
- An optional prefix “0x” or “0X” for hex values
- An optional prefix “0b” or “0B” for binary values
- An optional prefix “0” for octal values
- A string of digits
- An optional trailing string of tabs and spaces

The characters must match this generic format:

`[whitespace] [[sign] | [“0x”|“0X”] | [“0b”|“0B”]] [ddd] [whitespace]`

In this function, unrecognized character returns an error code.

Parameters

<code>pszString</code>	The string to convert.
<code>ul</code>	Points to the return buffer (an unsigned long).

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	If the function fails, the return value is zero.

Example

```

ULONG ul;

_atoul ("42", ul);

if (ul != 42)
    _Beep ();

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 90 of: 187
Project: GSEOS		

11.13 _dtoa ()

Prototype extern BOOL _dtoa (DOUBLE dValue,
 UCHAR &pszBuffer[]);

Description Converts a **double** number dValue to a null-terminated string and store the result in pszBuffer.

Note: The space allocated for the string must be large enough to hold the returned string, including the terminating null character (\0). The function can return up to 32 bytes.

Parameters

dValue	The double to convert.
pszBuffer	Points to the return buffer (an unsigned char array).

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function fails, the return value is zero.

Example

```

DOUBLE d;
UCHAR  aucBuffer[32];

d=42.0;
_dtoa (d, aucBuffer);
_MessageBox ("Test", aucBuffer, MB_OK | MB_ICONINFORMATION);

// returns: "4.200000e+01"

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 91 of: 187
Project: GSEOS		

11.14 _dtof32 ()

Prototype extern VOID _dtof32 (CONST DOUBLE &d, tsF32 &f32);

Description Converts a **double** number to a 32-bit IEEE float.

Parameters d The **double** to convert.

 f32 Points to the return buffer (a 32-bit IEEE float).

Return Value -

Example tsF32 f32;
 DOUBLE d;
 UCHAR szBuffer[1024];

 _dtof32 (3.45e-12, f32);
 _f32tod (f32, d);
 _dtoa (d, szBuffer);
 _MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

11.15 `_dtof32Array ()`

Prototype `extern BOOL _dtof32Array (CONST DOUBLE &ad[],
 tsF32 &af32[]);`

Description Converts an array of **double** numbers to an array of 32-bit IEEE floats.

Parameters `ad` The **double** array to convert.
`af32` Points to the return buffer (an array of 32-bit IEEE floats).

Return Value `TRUE` If the function succeeds, the return value is nonzero.
`FALSE` If the function fails, the return value is zero.

Example

```

tsF32  af32[3];
DOUBLE ad0[3], ad1[3];
UCHAR  szBuffer[1024];
ULONG  i;

ad0[0] = 3.45e-12;
ad0[1] = 0.0;
ad0[2] = 6.78e+30;

_dtof32Array (ad0, af32);
_f32todArray (af32, ad1);

for (i = 0; i < 3; ++i)
{
    _dtoa (ad1[i], szBuffer);
    _MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);
}

```

11.16 `_dtoi64 ()`

Prototype `extern VOID _dtoi64 (DOUBLE dValue,
 tsINT64 &i64);`

Description Converts a **double** number `dValue` to a 64-bit integer number `i64`.

Parameters `dValue` The **double** to convert.

 `i64` Points to the return buffer (a 64-bit integer).

Return Value -

Example

```
DOUBLE d;  
tsINT64 i64;  
  
d=-42.9;  
_dtoi64 (d, i64);  
if ((i64.ulLow != -42) || (i64.lHigh != -1))  
    _Beep ()
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 94 of: 187
Project: GSEOS		

11.17 `_f32tod ()`

Prototype `extern VOID _f32tod (CONST tsF32 &f32, DOUBLE &d);`

Description Converts a 32-bit IEEE float number to a **double**.

Parameters `f32` The 32-bit IEEE float to convert.

`d` Points to the return buffer (a **double**).

Return Value -

Example

```

tsF32 f32;
DOUBLE d;
UCHAR szBuffer[1024];

_dtof32 (3.45e-12, f32);
_f32tod (f32, d);
_dtoa (d, szBuffer);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 95 of: 187
Project: GSEOS		

11.18 `_f32todArray ()`

Prototype `extern BOOL _f32todArray (CONST tsF32 &af32[], DOUBLE &ad[]);`

Description Converts an array of 32-bit IEEE float numbers to a **double** array.

Parameters

<code>af32</code>	The 32-bit IEEE float array to convert.
<code>ad</code>	Points to the return buffer (a double array).

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	If the function fails, the return value is zero.

Example

```

tsF32  af32[3];
DOUBLE ad0[3], ad1[3];
UCHAR  szBuffer[1024];
ULONG  i;

ad0[0] = 3.45e-12;
ad0[1] = 0.0;
ad0[2] = 6.78e+30;

_dtof32Array (ad0, af32);
_f32todArray (af32, ad1);

for (i = 0; i < 3; ++i)
{
    _dtoa (ad1[i], szBuffer);
    _MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 96 of: 187
Project: GSEOS		

11.19 `_Gray16ToRgb32 ()`

Prototype extern BOOL `_Gray16ToRgb32` (VOID &paulRgbDestination,
 CONST VOID &ausGraySource,
 LONG lLow,
 LONG lHigh);

Description The function converts a 16-bit grayscale bitmap into a 32-bit RGB bitmap. In the destination bitmap all 3 color channels R, G, and B are set to the same value [0..255]. The simplified converting formula is:

$$\text{DestinationPixel}[i] = ((\text{SourcePixel}[i] - \text{lLow}) * 256) / (\text{lHigh} - \text{lLow}).$$

SourcePixel is assumed to be a 16-bit **unsigned** integer.

If lLow is equal lHigh the function returns an error. If lLow is greater than lHigh all pixels are inverted in brightness.

Parameters

<code>paulRgbDestination</code>	Points to destination array (array of ULONGs).
<code>ausGraySource</code>	Points to source array (array of USHORTs).
<code>lLow</code>	Black value in source (mapped to 0 in destination).
<code>lHigh</code>	White value in source (mapped to 255 in destination).

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function fails, the return value is zero.

Example

```
USHORT ausBuffer[1024];
ULONG  aulRgb[1024];
ULONG  ulElements, ulReturn, i;

ulElements = sizeof (ausBuffer) / sizeof (ausBuffer[0]);
if (_DlgGetUShortArray ("Enter Test Parameter.", ausBuffer,
                       ulElements, ulReturn))
{
    if (_Gray16ToRgb32 (aulRgb[0, ulReturn],
                      ausBuffer[0, ulReturn],
                      100, 1024))
        Scsiml355Write (aulRgb[0, ulReturn], TRUE);
    else
        _Beep ();
}
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 97 of: 187
Project: GSEOS		

11.20 `_GrayS16ToRgb32 ()`

Prototype `extern BOOL _GrayS16ToRgb32 (VOID &paulRgbDestination,
 CONST VOID &ausGraySource,
 LONG lLow,
 LONG lHigh);`

Description The function converts a 16-bit grayscale bitmap into a 32-bit RGB bitmap. In the destination bitmap all 3 color channels R, G, and B are set to the same value [0..255]. The simplified converting formula is:

$$\text{DestinationPixel}[i] = ((\text{SourcePixel}[i] - \text{lLow}) * 256) / (\text{lHigh} - \text{lLow}).$$

SourcePixel is assumed to be a 16-bit **signed** integer.

If lLow is equal lHigh the function returns an error. If lLow is greater than lHigh all pixels are inverted in brightness.

Parameters

<code>paulRgbDestination</code>	Points to destination array (array of ULONGs).
<code>ausGraySource</code>	Points to source array (array of USHORTs).
<code>lLow</code>	Black value in source (mapped to 0 in destination).
<code>lHigh</code>	White value in source (mapped to 255 in destination).

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	If the function fails, the return value is zero.

Example `// see Gray16ToRgb32()`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 98 of: 187
Project: GSEOS		

11.21 `_i64toa ()`

Prototype extern BOOL `_i64toa` (CONST tsINT64 &i64Value,
 UCHAR &pszBuffer[],
 ULONG ulRadix);

Description Converts a 64-bit integer number `i64Value` to a null-terminated string and store the result in `pszBuffer`.

`ulRadix` specifies the base to be used in converting value; it must be between 2 and 36, inclusive. If `i64Value` is negative and radix is 10, the first character of string is the minus sign (-).

Note: The space allocated for the string must be large enough to hold the returned string, including the terminating null character (0). The function can return up to 65 bytes.

Parameters	<code>i64Value</code>	The 64-bit integer to convert.
	<code>pszBuffer</code>	Points to the return buffer (an unsigned char array).
Return Value	TRUE	If the function succeeds, the return value is nonzero.
	FALSE	If the function fails, the return value is zero.

Example

```
tsINT64 i64;
UCHAR  szBuffer[80];

i64.ulLow = -42;
i64.lHigh = -1;

_i64toa (i64, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// returns: "-42"
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004
		Issue: 1.12 Date: 5/25/2004
Project: GSEOS		Page: 99 of: 187

11.22 `_i64tod ()`

Prototype extern VOID `_i64tod (CONST tsINT64 &i64Value,
 DOUBLE &d);`

Description Converts a 64-bit integer number `i64Value` to a **double** number.

Parameters `i64Value` The 64-bit integer to convert.

 `d` Points to the return buffer (a **double**).

Return Value -

Example `tsINT64 i64;
 DOUBLE d;

 i64.ulLow = -42;
 i64.lHigh = -1;

 _i64tod (i64, d);

 if (d != -42.0)
 _Beep ();`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 100 of: 187
Project: GSEOS		

11.23 `_ltoa ()`

Prototype extern BOOL `_ltoa` (LONG lValue,
 UCHAR &pszBuffer[],
 ULONG ulRadix);

Description Converts a **long** number lValue to a null-terminated string and store the result in pszBuffer.

ulRadix specifies the base to be used in converting value; it must be between 2 and 36, inclusive. If lValue is negative and radix equals 10, the first character of string is the minus sign (-).

Note: The space allocated for the string must be large enough to hold the returned string, including the terminating null character (\0). The function can return up to 33 bytes.

Parameters

lValue	The long to convert.
pszBuffer	Points to the return buffer (an unsigned char array).

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function fails, the return value is zero.

Example

```

LONG    l;
UCHAR  szBuffer[80];

l = -42;
_ltoa (l, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// returns: "-42"

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 101 of: 187
Project: GSEOS		

11.24 `_sprintf()`

Prototype extern LONG `_sprintf` (UCHAR &pszBuffer[],
CONST UCHAR &pszFormat[],
tssprintfParameter &p);

Description Writes formatted data to a string.
The function formats and stores a value in pszBuffer. The Parameter p is converted and output according to the corresponding format specification in pszFormat. The format consists of ordinary characters. A null character is appended after the last character written.

Note: According to stability of G-programs the `_sprintf` function differs slightly from the original C-Language `sprintf` function:

- 1) There is only one parameter for the value to format. If there are more than one format specifier, the value will be applied to each format specifier.
- 2) pszBuffer length will be checked
- 3) “%b” is supported (binary)
- 4) “%s” is not supported (string)
- 5) “%p” is not supported (pointer)

A format specification, which consists of optional and required fields, has the following form:

%[flags] [width] [.precision] [{h | l | I64 | L}]type

Each field of the format specification is a single character or a number signifying a particular format option. The simplest format specification contains only the percent sign and a *type* character (for example, %s). If a character that has no meaning as a format field follows a percent sign, the character is copied to pszBuffer. For example, to print a percent-sign character, use %%.

The optional fields, which appear before the *type* character, control other aspects of the formatting, as follows:

type

Required character that determines whether the value is interpreted as a character or a number

- c Character.
- d, i Signed decimal integer.
- u Unsigned decimal integer.
- o Unsigned octal integer.
- b Unsigned binary integer.
- x Unsigned hexadecimal integer, using “abcdef.”
- X Unsigned hexadecimal integer, using “ABCDEF.”
- e Signed value having the form [-]d.ddd e [sign]ddd where *d* is a single

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 102 of: 187
Project: GSEOS		

decimal digit, *dddd* is one or more decimal digits, *ddd* is exactly three decimal digits, and *sign* is + or –.

- E Identical to the **e** format except that **E** rather than **e** introduces the exponent.
- f Signed value having the form [–]*dddd.dddd*, where *dddd* is one or more decimal digits. The number of digits before the decimal point depends on the magnitude of the number, and the number of digits after the decimal point depends on the requested precision.
- g Signed value printed in **f** or **e** format, whichever is more compact for the given value and precision. The **e** format is used only when the exponent of the value is less than –4 or greater than or equal to the precision argument. Trailing zeros are truncated, and the decimal point appears only if one or more digits follow it.
- G Identical to the **g** format, except that **E**, rather than **e**, introduces the exponent (where appropriate).

flags

Optional character or characters that control justification of output and printing of signs, blanks, decimal points, and octal, binary, and hexadecimal prefixes. More than one flag can appear in a format specification.

- Left align the result within the given field width.
- + Prefix the output value with a sign (+ or –) if the output value is of a signed type.
- 0 If *width* is prefixed with **0**, zeros are added until the minimum width is reached. If 0 and – appear, the **0** is ignored. If **0** is specified with an integer format (**i**, **u**, **x**, **X**, **o**, **d**) the **0** is ignored.
- blank (' '), Prefix the output value with a blank if the output value is signed and positive; the blank is ignored if both the blank and + flags appear.
- # When used with the **o**, **x**, or **X** format, the # flag prefixes any nonzero output value with 0, 0x, or 0X, respectively.
 When used with the **e**, **E**, or **f** format, the # flag forces the output value to contain a decimal point in all cases.
 When used with the **g** or **G** format, the # flag forces the output value to contain a decimal point in all cases and prevents the truncation of trailing zeros.
 Ignored when used with **c**, **d**, **i**, or **u**.

width

Optional number that specifies the minimum number of characters output.

The *width* argument is a nonnegative decimal integer controlling the minimum number of characters printed. If the number of characters in the output value is less than the specified width, blanks are added to the left or the right of the values — depending on whether the – flag (for left alignment) is specified — until the minimum width is reached. If *width* is prefixed with 0, zeros are added until the mini-

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 103 of: 187
Project: GSEOS		

imum width is reached (not useful for left-aligned numbers).

The width specification never causes a value to be truncated. If the number of characters in the output value is greater than the specified width, or if *width* is not given, all characters of the value are printed.

precision

Optional number that specifies the maximum number of characters printed for all or part of the output field, or the minimum number of digits printed for integer values.

It specifies a nonnegative decimal integer, preceded by a period (.), which specifies the number of characters to be printed, the number of decimal places, or the number of significant digits. Unlike the width specification, the precision specification can cause either truncation of the output value or rounding of a floating-point value. If *precision* is specified as 0 and the value to be converted is 0, the result is no character output.

The type determines the interpretation of *precision* and the default when *precision* is omitted:

c	The precision has no effect.
d, i, u, o, b, x, X	The precision specifies the minimum number of digits to be printed. If the number of digits is less than <i>precision</i> , the output value is padded on the left with zeros. The value is not truncated when the number of digits exceeds <i>precision</i> . Default precision is 1.
e, E	The precision specifies the number of digits to be printed after the decimal point. The last printed digit is rounded. Default precision is 6; if <i>precision</i> is 0 or the period (.) appears without a number following it, no decimal point is printed.
f	The precision value specifies the number of digits after the decimal point. If a decimal point appears, at least one digit appears before it. The value is rounded to the appropriate number of digits. Default precision is 6; if <i>precision</i> is 0, or if the period (.) appears without a number following it, no decimal point is printed.
g, G	The precision specifies the maximum number of significant digits printed. Six significant digits are printed, with any trailing zeros truncated.

h | l | I64 | L

The optional prefixes to *type*, **h**, **l**, **I64**, and **L**, specify the “size” (long, short, character, depending upon the type specifier that they modify). These type-specifier prefixes are used with type characters in to specify interpretation of argument, as

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 104 of: 187
Project: GSEOS		

shown in the following table.

- h Signed short int when used with d, i, o, b, x, X.
Unsigned short int, used with u.
- l Signed long int, used with d, i, o, b, x, X.
Unsigned long int, used with u.
- I64 Signed 64-bit integer, used with d, i, o, b, x, X.
Unsigned 64-bit integer, used with u.
- L Long double, used with e, E, f, g, G.

Parameters

pszBuffer Storage location for output.

pszFormat Format-control string.

p Contains the value to format.

```
// _sprintf parameter
typedef union
{
    LONG      l;
    ULONG     ul;
    DOUBLE    d;
    tsINT64   i64;
    tsUINT64  ui64;
} tssprintfParameter;
```

Return Value On success, the function returns the number of bytes in pszBuffer. The return value does not include the terminating null byte in the count. On error, the function returns -1.

Example

```
UCHAR szBuffer[1024];
tssprintfParameter p;

p.l = 0x12345;
_sprintf (szBuffer, "%08X", p);           // 00012345

p.l = -42;
_sprintf (szBuffer, "%d", p);            // -42

p.d = 1.234e-2;
_sprintf (szBuffer, "%f", p);           // 0.012340

p.i64.ulLow = 0x89abcdef;
p.i64.lHigh = 0x01234567;
_sprintf (szBuffer, "0x%016I64X", p);    // 0x0123456789ABCDEF
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 105 of: 187
Project: GSEOS		

11.25 `_ToIntel16 ()`

Prototype `extern USHORT _ToIntel16 (USHORT us);`

Description Converts an **unsigned short** number from default G-Language format into INTEL format.

Parameters `us` The **unsigned short** to convert.

Return Value On INTEL machines the function returns the unmodified value.
 On MOTOROLA machines the function returns the swapped value.

Example `USHORT us;`
`us = _ToIntel16 (0x1234);`
`// on INTEL machines: us == 0x1234`
`// on MOTOROLA machines: us == 0x3412`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 106 of: 187
Project: GSEOS		

11.26 `_ToIntel32 ()`

Prototype `extern ULONG _ToIntel32 (ULONG ul);`

Description Converts an **unsigned long** number from default G-Language format into INTEL format.

Parameters `ul` The **unsigned long** to convert.

Return Value On INTEL machines the function returns the unmodified value.
On MOTOROLA machines the function returns the swapped value.

Example `ULONG ul;`

`ul = _ToIntel32 (0x12345678);`

`// on INTEL machines: ul == 0x12345678`
`// on MOTOROLA machines: ul == 0x78563412`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 107 of: 187
Project: GSEOS		

11.27 `_ToIntel64 ()`

Prototype extern VOID `_ToIntel64` (CONST tsUINT64 &ui64,
 tsUINT64 &ui64Result);

Description The function converts an unsigned 64-bit integer number from default G-Language format into INTEL format.

The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer structure for both input and result.

On INTEL machines the function copies the unmodified value.

On MOTOROLA machines the function swaps and copies the value.

Parameters i64 The first parameter.

 i64Result The result of the calculation.

Return Value -

Example

```
tsUINT64 ui64, ui64Result;
UCHAR    szBuffer[128];

ui64.ulLow  = 0x76543210;
ui64.ulHigh = 0xFEDCBA98;

_ToIntel64 (ui64, ui64Result);
_ui64toa (ui64Result, szBuffer, 16);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result
// on INTEL machines:    0xfedcba9876543210
// on MOTOROLA machines: 0x1032547698badcfe
```

IDA Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 108 of: 187
Project: GSEOS		

11.28 `_ToIntel16Array ()`

Prototype extern VOID `_ToIntel16Array` (CONST USHORT &ausSource[],
 USHORT &ausDestination[]);

Description Converts an array of **unsigned short** numbers from default G-Language format into INTEL format. Even when the source and destination arrays overlap, the overlapping locations are converted and copied correctly.

On INTEL machines the function copies unmodified values.

On MOTOROLA machines the function copies swapped values.

Parameters ausSource The array to convert.

 ausDestination The converted array.

Return Value -

Example

```
USHORT aus1[3], aus2[3];
UCHAR  szBuffer[80];
ULONG  i;

aus1[0] = 0x0123;
aus1[1] = 0x4567;
aus1[2] = 0x89AB;

_ToIntel16Array (aus1, aus2);

for (i = 0; i < sizeof (aus2) / sizeof (aus2[0]); ++i)
{
    _ultoa (aus2[i], szBuffer, 16);
    _AddMessage (MSG_INFORMATION, "Test", szBuffer);
}

// result
// on INTEL machines: 0x0123, 0x4567, 0x89AB
// on MOTOROLA machines: 0x2301, 0x6745, 0xAB89
```


IDA Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG <hr/> Project: GSEOS	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 110 of: 187
--	---	--

11.30 `_ToIntel64Array ()`

Prototype extern VOID `_ToIntel64Array` (CONST tsUINT64 &ai64Source[],
 tsUINT64 &ai64Destination[]);

Description Converts an array of unsigned 64-bit integer numbers from default G-Language
 format into INTEL format. Even when the source and destination arrays overlap,
 the overlapping locations are converted and copied correctly.

On INTEL machines the function copies unmodified values.

On MOTOROLA machines the function copies swapped values.

Parameters ai64Source The array to convert.

 ai64Destination The converted array.

Return Value -

Example

```

tsUINT64 ai64a[2], ai64b[2];
UCHAR szBuffer[80];
ULONG i;

ai64a[0].ulHigh = 0x01234567;
ai64a[0].ulLow = 0x89ABCDEF;
ai64a[1].ulHigh = 0x42434445;
ai64a[1].ulLow = 0x46474849;

_ToIntel64Array (ai64a, ai64b);

for (i = 0; i < sizeof (ai64b) / sizeof (ai64b[0]); ++i)
{
  _ui64toa (ai64b[i], szBuffer, 16);
  _AddMessage (MSG_INFORMATION, "Test", szBuffer);
}

// result
// on INTEL machines: 0x0123456789ABCDEF, 0x4243444546474849
// on MOTOROLA machines: 0xEFCDAB8967452301, 0x4948474645444342

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 111 of: 187
Project: GSEOS		

11.31 `_ToMotorola16 ()`

Prototype `extern USHORT _ToMotorola16 (USHORT us);`

Description Converts an **unsigned short** number from default G-Language format into MOTOROLA format.

Parameters `us` The **unsigned short** to convert.

Return Value On INTEL machines the function returns the swapped value.
On MOTOROLA machines the function returns the unmodified value.

Example

```
USHORT us;

us = _ToMotorola16 (0x1234);

// on INTEL machines:    us == 0x3412
// on MOTOROLA machines: us == 0x1234
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 112 of: 187
Project: GSEOS		

11.32 `_ToMotorola32 ()`

Prototype `extern ULONG _ToMotorola32 (ULONG ul);`

Description Converts an **unsigned long** number from default G-Language format into MOTOROLA format.

Parameters `ul` The **unsigned long** to convert.

Return Value On INTEL machines the function returns the swapped value.
On MOTOROLA machines the function returns the unmodified value.

Example `ULONG ul;`

`ul = _ToMotorola32 (0x12345678);`

`// on INTEL machines: ul == 0x78563412`
`// on MOTOROLA machines: ul == 0x12345678`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 113 of: 187
Project: GSEOS		

11.33 `_ToMotorola64 ()`

Prototype extern VOID `_ToMotorola64` (CONST tsUINT64 &ui64,
 tsUINT64 &ui64Result);

Description The function converts an unsigned 64-bit integer number from default G-Language format into MOTOROLA format.

The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer structure for both input and result.

On INTEL machines the function swaps and copies the value.

On MOTOROLA machines the function copies the unmodified value.

Parameters i64 The first parameter.

 i64Result The result of the calculation.

Return Value -

Example

```

tsUINT64 ui64, ui64Result;
UCHAR    szBuffer[128];

ui64.ulLow  = 0x76543210;
ui64.ulHigh = 0xFEDCBA98;

_ToMotorola64 (ui64, ui64Result);
_ui64toa (ui64Result, szBuffer, 16);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result
// on INTEL machines:    0x1032547698badcfe
// on MOTOROLA machines: 0xfedcba9876543210

```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 115 of: 187
Project: GSEOS		

11.35 `_ToMotorola32Array ()`

Prototype `extern VOID _ToMotorola32Array (CONST ULONG &aulSource[],
ULONG &aulDestination[]);`

Description Converts an array of **unsigned long** numbers from default G-Language format into MOTOROLA format. Even when the source and destination arrays overlap, the overlapping locations are converted and copied correctly.
On INTEL machines the function copies swapped values.
On MOTOROLA machines the function copies unmodified values.

Parameters `aulSource` The array to convert.

`aulDestination` The converted array.

Return Value -

Example

```

ULONG aul1[2], aul2[2];
UCHAR szBuffer[80];
ULONG i;

aul1[0] = 0x01234567;
aul1[1] = 0x89ABCDEF;

_ToMotorola32Array (aul1, aul2);

for (i = 0; i < sizeof (aul2) / sizeof (aul2[0]); ++i)
{
    _ultoa (aul2[i], szBuffer, 16);
    _AddMessage (MSG_INFORMATION, "Test", szBuffer);
}

// result
// on INTEL machines:      0x67452301, 0xEFCDAB89
// on MOTOROLA machines: 0x01234567, 0x89ABCDEF

```

11.36 `_ToMotorola64Array ()`

Prototype `extern VOID _ToMotorola64Array (CONST tsUINT64 &ai64Source[],
 tsUINT64 &ai64Destination[]);`

Description Converts an array of unsigned 64-bit integer numbers from default G-Language format into MOTOROLA format. Even when the source and destination arrays overlap, the overlapping locations are converted and copied correctly.

On INTEL machines the function copies swapped values.

On MOTOROLA machines the function copies unmodified values.

Parameters `ai64Source` The array to convert.

`ai64Destination` The converted array.

Return Value -

Example

```
tsUINT64  ai64a[2], ai64b[2];
UCHAR     szBuffer[80];
ULONG     i;

ai64a[0].ulHigh = 0x01234567;
ai64a[0].ulLow  = 0x89ABCDEF;
ai64a[1].ulHigh = 0x42434445;
ai64a[1].ulLow  = 0x46474849;

_ToMotorola64Array (ai64a, ai64b);

for (i = 0; i < sizeof (ai64b) / sizeof (ai64b[0]); ++i)
{
  _ui64toa (ai64b[i], szBuffer, 16);
  _AddMessage (MSG_INFORMATION, "Test", szBuffer);
}

// result
// on INTEL machines:      0xEFCDAB8967452301, 0x4948474645444342
// on MOTOROLA machines:  0x0123456789ABCDEF, 0x4243444546474849
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 117 of: 187
Project: GSEOS		

11.37 `_ui64toa ()`

Prototype extern BOOL `_ui64toa` (CONST tsUINT64 &ui64Value,
 UCHAR &pszBuffer[],
 ULONG ulRadix);

Description Converts an unsigned 64-bit integer number `ui64Value` to a null-terminated string and store the result in `pszBuffer`.

`ulRadix` specifies the base to be used in converting value; it must be between 2 and 36, inclusive. The function does not set a minus sign (-) in the converted string.

Note: The space allocated for string must be large enough to hold the returned string, including the terminating null character (0). The function can return up to 65 bytes.

Parameters

<code>ui64Value</code>	The unsigned 64-bit integer to convert.
<code>pszBuffer</code>	Points to the return buffer (an unsigned char array).

Return Value

TRUE	If the function succeeds, the return value is nonzero.
FALSE	If the function fails, the return value is zero.

Example

```
tsUINT64 ui64;
UCHAR    szBuffer[80];

ui64.ulLow = 42;
ui64.ulHigh = 0;

_ui64toa (ui64, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// returns: "42"
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 118 of: 187
Project: GSEOS		

11.38 _ultoa ()

Prototype extern BOOL _ultoa (ULONG ulValue,
 UCHAR &pszBuffer[],
 ULONG ulRadix);

Description Converts an **unsigned long** number ulValue to a null-terminated string and store the result in pszBuffer.

ulRadix specifies the base to be used in converting value; it must be between 2 and 36, inclusive. The function does not set a minus sign (-) in the converted string.

Note: The space allocated for string must be large enough to hold the returned string, including the terminating null character (\0). The function can return up to 33 bytes.

Parameters	ulValue	The unsigned long to convert.
	pszBuffer	Points to the return buffer (an unsigned char array).
Return Value	TRUE	If the function succeeds, the return value is nonzero.
	FALSE	If the function fails, the return value is zero.

Example

```

ULONG ul;
UCHAR szBuffer[80];

ul = -42;

_ultoa (ul, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// returns: "4294967254"

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1 style="margin: 0;">GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 119 of: 187
Project: GSEOS		

12 Time Functions

The GSEOS provides several date and time functions and structures to retrieve and set the date and time for the system. This section describes various functions and structures and explains how to use them to examine and modify dates and times.

System time is the current date and time of day. Windows keeps system time so that your applications have ready access to accurate time. When Windows first starts, it sets the system time to a value based on the real-time clock of the computer and thereafter regularly updates the time.

Windows bases system time on coordinated universal time (UTC). UTC-based time is loosely defined as the current date and time of day in Greenwich, England.

The GSEOS supports three different time formats:

- system time as structure:

The structure `tsSystemTime` represents a date and time using individual members for the month, day, year, weekday, hour, minute, second, and millisecond.

```
typedef struct
{
    USHORT usYear;           // year
    USHORT usMonth;         // month; January = 1, February = 2, ...
    USHORT usDayOfWeek;     // day of the week; Sunday = 0, Monday = 1, ...
    USHORT usDay;           // day of the month
    USHORT usHour;          // hour
    USHORT usMinute;        // minute
    USHORT usSecond;        // second
    USHORT usMilliseconds;  // millisecond
} tsSystemTime;
```

It is not recommended that you add and subtract values from the `tsSystemTime` structure to obtain relative times. Instead, you should convert the `tsSystemTime` structure to a 64-bit integer via `_SystemTimeToi64 ()` and use normal 64-bit arithmetic on the 64-bit value.

- system time as 64-bit integer

The system time as 64-bit integer value based on the time format builds in Windows NT and representing the number of 100-nanosecond intervals since 00:00:00 on January 1, 1601 (UTC). It is recommended to use a 64-bit time format to handle relative times.

- Unix time as 32-bit integer

For compatibility GSEOS supports the Unix time format. Unix time represents the number of seconds since 00:00:00 on January 1, 1970 (UTC). Note the limited range of the Unix time format: the value is valid up to 03:14:07 on January 19, 2038.

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 120 of: 187
Project: GSEOS		

12.1 _GetSystemTime ()

Prototype `extern BOOL _GetSystemTime (tsSystemTime &sSystemTime);`

Description The function retrieves the current system date and time. The system time is expressed in Coordinated Universal Time (UTC).

Parameters `sSystemTime` Structure to receive the current system date and time.

Return Value `TRUE` If the function succeeds, the return value is nonzero.

`FALSE` If the function fails, the return value is zero.

Example

```

tsSystemTime s;
UCHAR        szBuffer[80];

_GetSystemTime (s);
_SystemTimeToString (s, szBuffer, 0);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 121 of: 187
Project: GSEOS		

12.2 _GetSystemTimei64 ()

Prototype extern BOOL _GetSystemTimei64 (tsINT64 &i64);

Description The function retrieves the current system date and time.
The system time is expressed in a 64-bit integer representing the number of 100-nanosecond intervals since 00:00:00 on January 1, 1601 (UTC). It is recommended to use a 64-bit time format to handle relative times.

Parameters i64 64-bit integer to receive the current system date and time.

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If the function fails, the return value is zero.

Example

```

tsINT64 i64Start, i64Stop, i64Diff, i64;
UCHAR  szBuffer[80];

// stopwatch
_GetSystemTimei64 (i64Start);
seed ();
sleep (random (1000));
_GetSystemTimei64 (i64Stop);

// calculate interval
_i64Sub (i64Stop, i64Start, i64Diff);

// convert 100ns ticks to ms
i64.ulLow = 10000;
i64.lHigh = 0;
_i64Div (i64Diff, i64, i64Diff);

// display interval
_i64toa (i64Diff, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 122 of: 187
Project: GSEOS		

12.3 _GetUnixTime ()

Prototype extern ULONG _GetUnixTime ();

Description The function retrieves the current system date and time.
 Unix time represents the number of seconds since 00:00:00 on January 1, 1970 (UTC).

Note: The Unix time is valid up to 03:14:07 on January 19, 2038.

Parameters -

Return Value Number of seconds since 00:00:00 on January 1, 1970 (UTC).

Example

```

ULONG ul;
UCHAR szBuffer[80];

ul = _GetUnixTime ();

_ultoa (ul, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 123 of: 187
Project: GSEOS		

12.4 _i64SystemTimeToString ()

Prototype extern BOOL _i64SystemTimeToString (CONST tsINT64 &i64,
 UCHAR &pszBuffer[],
 ULONG ulFormat);

Description The function converts a 64-bit integer time (counting 100ns intervals) to a null-terminated character string. It is recommended to use a 64-bit time format to handle relative times.

Note: The return buffer must have at least a length of 32.

Parameters

i64	Containing a 64-bit time to convert.
pszBuffer	Character array to receive the string.
ulFormat	Specifies the string format:
	0: <D,>hh:mm:ss.nnn
	1: <D,>hh:mm:ss
	DD days (the field will be skipped if days are zero)
	hh hour
	mm minute
	ss second
	nnn millisecond

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If the function fails, the return value is zero.

Example

```
tsINT64 i64Start, i64Stop, i64Diff;
UCHAR  szBuffer[80];
ULONG  i;

_GetSystemTimei64 (i64Start);
seed();
sleep (random (2000));
_GetSystemTimei64 (i64Stop);

_i64Sub (i64Stop, i64Start, i64Diff);

for (i = 0; i < 2; ++i)
{
  _i64SystemTimeToString (i64Diff, szBuffer, i);
  _AddMessage (MSG_INFORMATION, "Test", szBuffer);
}

// results:
// "00:00:01.828"
// "00:00:01"
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 126 of: 187
Project: GSEOS		

12.7 _RPROGetCurrentTime ()

Prototype extern BOOL _RPROGetCurrentTime (UCHAR &aucBuffer[]);

Description The function retrieves the current system date and time. The system time is expressed in Coordinated Universal Time (UTC).

The date and time is stored in a special format for the ROSETTA network protocol: “YYYY,DDD,hh,mm,ss,nnn<space>” (e.g. “2000,152,12,55,03,997 ”)

 YYYY year
 DDD day of year
 hh hour
 mm minute
 ss second
 nnn millisecond

Note: The length of the sting is exactly 22 bytes and is not terminated by a null character.

Parameters aucBuffer Character array to receive the current system date and time.

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If the function fails, the return value is zero.

Example

```
UCHAR szBuffer[80];

_RPROGetCurrentTime (szBuffer);
szBuffer[22] = '\0'; // RPRO time is not null-terminated!

_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 127 of: 187
Project: GSEOS		

12.8 _RPROSetCurrentTime ()

Prototype extern BOOL _RPROSetCurrentTime (CONST UCHAR &aucBuffer[]);

Description The function sets the current system date and time. The system time is expressed in Coordinated Universal Time (UTC).

The date and time is stored in a special format for the ROSETTA network protocol: “YYYY,DDD,hh,mm,ss,nnn<space>” (e.g. “2000,152,12,55,03,997 ”)

YYYY year
DDD day of year
hh hour
mm minute
ss second
nnn millisecond

Note 1: The length of the sting is exactly 22 bytes and is not terminated by a null character.

Note 2: The function fails if the GSEOS program does not have the privilege to set the system time (User Rights Policy: “Change the system time”). This privilege is disabled by default for normal user and enabled by default for administrators.

Parameters aucBuffer Character array of system date and time to set.

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If the function fails, the return value is zero.

Example

```

UCHAR aucRPROTime[22];
ULONG ulYear;
UCHAR szBuffer[80];

_RPROGetCurrentTime (aucRPROTime);
szBuffer[0..3] = aucRPROTime[0..3];
szBuffer[4] = '\0';
_atoul (szBuffer, ulYear);
--ulYear;
_ultoa (ulYear, szBuffer, 10);
aucRPROTime[0..3] = szBuffer[0..3];
_RPROSetCurrentTime (aucRPROTime);

_MessageBox ("Test", "System time manipulated.",
             MB_OK | MB_ICONINFORMATION);

_RPROGetCurrentTime (aucRPROTime);
++ulYear;
_ultoa (ulYear, szBuffer, 10);
aucRPROTime[0..3] = szBuffer[0..3];
_RPROSetCurrentTime (aucRPROTime);

_MessageBox ("Test", "System time restored.",
             MB_OK | MB_ICONINFORMATION);

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 128 of: 187
Project: GSEOS		

12.9 _SetSystemTime ()

Prototype `extern BOOL _SetSystemTime (CONST tsSystemTime &sSystemTime);`

Description The function sets the current system date and time. The system time is expressed in Coordinated Universal Time (UTC).

Note: The function fails if the GSEOS program does not have the privilege to set the system time (User Rights Policy: “Change the system time”). This privilege is disabled by default for normal user and enabled by default for administrators.

Parameters `sSystemTime` Contains the system date and time to set.

Return Value `TRUE` If the function succeeds, the return value is nonzero.

`FALSE` If the function fails, the return value is zero.

Example `tsSystemTime sSystemTime;`

```

_GetSystemTime (sSystemTime);
++sSystemTime.usYear;
_SetSystemTime (sSystemTime);

_MessageBox ("Test", "System time manipulated.",
             MB_OK | MB_ICONINFORMATION);

_GetSystemTime (sSystemTime);
--sSystemTime.usYear;
_SetSystemTime (sSystemTime);

_MessageBox ("Test", "System time restored.",
             MB_OK | MB_ICONINFORMATION);

```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 129 of: 187
Project: GSEOS		

12.10 `_SetSystemTimei64 ()`

Prototype `extern BOOL _SetSystemTimei64 (CONST tsINT64 &i64);`

Description The function sets the current system date and time. The system time representing the number of 100-nanosecond intervals since 00:00:00 on January 1, 1601 (UTC).

Note: The function fails if the GSEOS program does not have the privilege to set the system time (User Rights Policy: “Change the system time”). This privilege is disabled by default for normal user and enabled by default for administrators.

Parameters `i64` Contains the system date and time to set.

Return Value `TRUE` If the function succeeds, the return value is nonzero.

`FALSE` If the function fails, the return value is zero.

Example

```
tsINT64 i64;

_GetSystemTimei64 (i64);
i64.lHigh += 10000;
_SetSystemTimei64 (i64);

_MessageBox ("Test", "System time manipulated.",
             MB_OK | MB_ICONINFORMATION);

_GetSystemTimei64 (i64);
i64.lHigh -= 10000;
_SetSystemTimei64 (i64);

_MessageBox ("Test", "System time restored.",
             MB_OK | MB_ICONINFORMATION);
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 130 of: 187
Project: GSEOS		

12.11 `_SystemTimeToi64 ()`

Prototype `extern BOOL _SystemTimeToi64 (CONST tsSystemTime &sSystemTime, tsINT64 &i64);`

Description The function converts a system time structure to a 64-bit integer representing the number of 100-nanosecond intervals since 00:00:00 on January 1, 1601 (UTC).

Parameters

<code>sSystemTime</code>	Containing the system date and time to convert.
<code>i64</code>	Structure to receive the converted 64-bit system time.

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	If the function fails, the return value is zero.

Example

```
tsSystemTime sSystemTime;
tsINT64      i64;
UCHAR       szBuffer[80];

_GetSystemTime (sSystemTime);
_SystemTimeToi64 (sSystemTime, i64);

_i64toa (i64, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 131 of: 187
Project: GSEOS		

12.12 `_SystemTimeToLocalTime ()`

Prototype `extern BOOL _SystemTimeToLocalTime (`
`CONST tsSystemTime &sSystemTime,`
`tsSystemTime &sLocalTime);`

Description The function converts a system time (in UTC) to a local time (corresponding to current time zone settings).

Parameters

<code>sSystemTime</code>	<code>System time to convert.</code>
<code>sLocalTime</code>	<code>Structure to receive the converted local time.</code>

Return Value

<code>TRUE</code>	<code>If the function succeeds, the return value is nonzero.</code>
<code>FALSE</code>	<code>If the function fails, the return value is zero.</code>

Example

```
tsSystemTime sUTC, sLocal, st;
UCHAR       szBuffer[1024];

_GetSystemTime (sUTC);
_SystemTimeToString (sUTC, szBuffer, 0);
_AddMessage (MSG_INFORMATION, "Test", szBuffer);

_SystemTimeToLocalTime (sUTC, sLocal);

_SystemTimeToString (sLocal, szBuffer, 0);
_AddMessage (MSG_INFORMATION, "Test", szBuffer);

_LocalTimeToSystemTime (sLocal, st);

_SystemTimeToString (st, szBuffer, 0);
_AddMessage (MSG_INFORMATION, "Test", szBuffer);
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 132 of: 187
Project: GSEOS		

12.13 `_SystemTimeToString ()`

Prototype extern BOOL `_SystemTimeToString` (CONST tsSystemTime &sSystemTime,
 UCHAR &pszBuffer[],
 ULONG ulFormat);

Description The function converts a system time structure to null-terminated character string.
Note: The return buffer must have at least a length of 32.

Parameters sSystemTime Containing the system date and time to convert.

 pszBuffer Structure to receive the string.

 ulFormat Specifies the string format:

- 0: DD/MM/YYYY hh:mm:ss.nnn
- 1: DD/MM/YYYY hh:mm:ss
- 2: hh:mm:ss.nnn
- 3: hh:mm:ss
- 4: YYYYMMDD_hhmmss
- 5: WWW MMM DD hh:mm:ss YYYY

DD day of month
MM month
YYYY year
hh hour
mm minute
ss second
nnn millisecond
WWW day of week (3 characters)
MMM month (3 characters)

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If the function fails, the return value is zero.

Example

```

tsSystemTime sSystemTime;
UCHAR       szBuffer[80];
ULONG       i;

_GetSystemTime (sSystemTime);

for (i = 0; i < 6; ++i)
{
    _SystemTimeToString (sSystemTime, szBuffer, i);
    _AddMessage (MSG_INFORMATION, "Test", szBuffer);
}

// results:
// 0: "06/07/2000 09:27:12.906"

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 133 of: 187

```
// 1: "06/07/2000 09:27:12"  
// 2: "09:27:12.906"  
// 3: "09:27:12"  
// 4: "20000607_092712"  
// 5: "Wed Jun 07 09:27:12 2000"
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 134 of: 187
Project: GSEOS		

12.14 `_SystemTimeToUnixTime ()`

Prototype `extern BOOL _SystemTimeToUnixTime (`
 `CONST tsSystemTime &sSystemTime,`
 `ULONG &ulTime);`

Description The function converts a system time structure to Unix time representing the number of seconds since 00:00:00 on January 1, 1970 (UTC).

Note: The Unix time is valid up to 03:14:07 on January 19, 2038.

Parameters

<code>sSystemTime</code>	Containing the system date and time to convert.
<code>ulTime</code>	Structure to receive the converted 32-bit Unix time.

Return Value

<code>TRUE</code>	If the function succeeds, the return value is nonzero.
<code>FALSE</code>	If the function fails, the return value is zero.

Example

```
tsSystemTime sSystemTime;
ULONG        ulTime;
UCHAR        szBuffer[80];

_GetSystemTime (sSystemTime);
_SystemTimeToUnixTime (sSystemTime, ulTime);

_ultoa (ulTime, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);
```

IDA Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004
		Issue: 1.12 Date: 5/25/2004
Project: GSEOS		Page: 135 of: 187

12.15 `_UnixTimeToSystemTime ()`

Prototype `extern BOOL _UnixTimeToSystemTime (ULONG ulUnixTime,
tsSystemTime &sSystemTime);`

Description The function converts a Unix time representing the number of seconds since 00:00:00 on January 1, 1970 (UTC) to a system time structure.

Note: The Unix time is valid up to 03:14:07 on January 19, 2038.

Parameters `ulTime` Containing the 32-bit Unix time to convert.

`sSystemTime` Structure to receive the converted system date and time.

Return Value `TRUE` If the function succeeds, the return value is nonzero.

`FALSE` If the function fails, the return value is zero.

Example

```
tsSystemTime sSystemTime;  
ULONG        ulTime;  
UCHAR        szBuffer[80];  
  
ulTime = _GetUnixTime ();  
_UnixTimeToSystemTime (ulTime, sSystemTime);  
  
_ultoa (sSystemTime.usYear, szBuffer, 10);  
_MessageBox ("Current Year is:", szBuffer,  
            MB_OK | MB_ICONINFORMATION);
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 136 of: 187

13 String Functions

The GSEOS provides several string-manipulation and memory-manipulation routines. Stings are null-terminated character strings inside an unsigned character array.

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 137 of: 187
Project: GSEOS		

13.1 _memcmp ()

Prototype extern LONG _memcmp (CONST VOID &pauc1,
 CONST VOID &pauc2);

Description Compares two memory areas.

The function performs an unsigned comparison of pauc1 to pauc2, starting with the first character in each area and continuing with subsequent characters until the corresponding characters differ or until the end of one memory area is reached.

Note: Different behavior than the corresponding C-Function! The G-Language knows the length of both memory areas. So the function needs no length parameter and stops comparing at the end of the memory area automatically.

Parameters psz1 String 1.
 psz2 String 2.

Return Value < 0 If pauc1 is less than pauc2.
 = 0 If pauc1 the same as pauc2.
 > 0 If pauc1 greater than pauc2.

Example

```

struct
{
    DOUBLE d;
    LONG   l;
} s1, s2;
LONG l;

s1.d = 1.0;
s1.l = -1;
s2.d = 1.0;
s2.l = -1;

l = _memcmp (s1, s2);
if (l == 0)
    _Beep ();

// you may also compare directly!!!
if (s1 == s2)
    _Beep ();
  
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 138 of: 187
Project: GSEOS		

13.2 _memmove ()

Prototype extern VOID _memmove (CONST VOID &pvDest,
 CONST VOID &pvSrc);

Description Moves one buffer to another.

The function copies from pvSrc to pvDest. If some regions of the source area and the destination overlap, _memmove ensures that the original source bytes in the overlapping region are copied before being overwritten.

Note: Different behavior than the corresponding C-Function! The G-Language knows the length of both memory areas. So the function needs no length parameter and stops at the minimum buffer length automatically.

Parameters pvDest Destination buffer.

 pvSrc Source buffer.

Return Value -

Example

```

struct
{
    DOUBLE d;
    LONG   l;
} s1, s2;
LONG l;

s1.d = 1.0;
s1.l = -1;
_memmove (s2, s1);

l = _memcmp (s1, s2);
if (l == 0)
    _Beep ();

// you may also compare directly!!!
if (s1 == s2)
    _Beep ();

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 139 of: 187
Project: GSEOS		

13.3 `_memset ()`

Prototype extern VOID `_memset` (VOID &pauc,
 UCHAR ucFill,
 ULONG ulLength);

Description Sets buffers to a specified character.
 The function sets the first ulLength bytes of pauc to the character ucFill.

Parameters

pauc	Pointer to buffer.
ucFill	Character to set.
ulLength	Number of characters.

Return Value -

Example

```

UCHAR szBuffer[32];

_memset (szBuffer, 'x', 32);
szBuffer[31] = '\0';
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 140 of: 187
Project: GSEOS		

13.4 _strcat ()

Prototype extern VOID _strcat (UCHAR &pszDest[],
 CONST UCHAR &pszScr[]);

Description Appends one string to another.

 The function appends pszScr to pszDest and terminates the resulting string with a null character. The initial character of pszScr overwrites the terminating null character of pszDest. The length of the resulting string is _strlen(pszDest) + _strlen(pszScr).

Overflow checking is performed when strings are copied or appended. The behaviour of the function is undefined if the source and destination strings overlap.

Parameters pszDest Destination string.

 pszScr String to append.

Return Value -

Example UCHAR sz1[80];

 _strcpy (sz1, "123");
 _strcat (sz1, "456");
 _MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);

 // result: "123456"

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 141 of: 187
Project: GSEOS		

13.5 `_strcmp ()`

Prototype extern LONG `_strcmp` (CONST UCHAR &psz1[],
 CONST UCHAR &psz2[]);

Description Compares one string to another.

The function performs an unsigned comparison of psz1 to psz2, starting with the first character in each string and continuing with subsequent characters until the corresponding characters differ or until the end of the strings is reached.

Parameters psz1 String 1.

 psz2 String 2.

Return Value < 0 If psz1 is less than psz2.

 = 0 If psz1 the same as psz2.

 > 0 If psz1 greater than psz2.

Example LONG l;

```
l = _strcmp ("abc", "abd");
if (l < 0)
    _Beep ();
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 142 of: 187
Project: GSEOS		

13.6 `_strcmpi ()`

Prototype extern LONG `_strcmpi` (CONST UCHAR &psz1[],
 CONST UCHAR &psz2[]);

Description Compares one string to another, without case sensitivity.
The function performs an unsigned comparison of psz1 to psz2, without case sensitivity, starting with the first character in each string and continuing with subsequent characters until the corresponding characters differ or until the end of the strings is reached.

Parameters

psz1	String 1.
psz2	String 2.

Return Value

< 0	If psz1 is less than psz2.
= 0	If psz1 the same as psz2.
> 0	If psz1 greater than psz2.

Example

```

LONG l;

l = _strcmpi ("Abc", "abc");
if (l == 0)
    _Beep ();

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 143 of: 187
Project: GSEOS		

13.7 `_strcpy ()`

Prototype extern VOID `_strcpy` (UCHAR &pszDest[],
 CONST UCHAR &pszScr[]);

Description Copies one string into another.
 The function copies pszScr, including the terminating null character, to the location specified by pszDest.
 Overflow checking is performed when strings are copied or appended. The behaviour of the function is undefined if the source and destination strings overlap.

Parameters

pszDest	Destination string.
pszScr	String to copy.

Return Value -

Example

```

UCHAR sz1[80];

_strcpy (sz1, "123");
_strcat (sz1, "456");
_MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);

// result: "123456"

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 144 of: 187
Project: GSEOS		

13.8 _strlen ()

Prototype extern ULONG _strlen (CONST UCHAR &psz[]);

Description Get the length of a string.
 The function returns the number of characters in string psz, not including the terminating null character.

Parameters psz The string.

Return Value Length of string excluding the terminating null character.

Example

```

ULONG ul;
UCHAR szBuffer[128];

ul = _strlen ("123");
_ultoa (ul, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: 3
  
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 145 of: 187
Project: GSEOS		

13.9 _strlwr ()

Prototype extern VOID _strlwr (UCHAR &psz[]);

Description Converts uppercase letters in a string to lowercase.
 The function converts uppercase letters (A to Z) to lowercase letters (a to z). The modification is done in place. No other characters are changed.

Parameters psz String to convert.

Return Value -

Example UCHAR sz1[80];
 _strcpy (sz1, "AbCdE123");
 _strlwr (sz1);
 _MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);
 // result: "abcde123"

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 146 of: 187
Project: GSEOS		

13.10 `_strncat()`

Prototype extern VOID `_strncat` (UCHAR &pszDest[],
 CONST UCHAR &pszScr[],
 ULONG ulMaxLen);

Description Appends one string to another.
The function appends, at most, the first `ulMaxLen` characters of `pszScr` to `pszDest`. The initial character of `pszScr` overwrites the terminating null character of `pszDest`. If a null character appears in `pszScr` before `ulMaxLen` characters are appended, the function appends all characters from `pszScr`, up to the null character. The resulting string is terminated with a null character. Overflow checking is performed when strings are copied or appended. The behaviour of the function is undefined if the source and destination strings overlap.

Parameters

<code>pszDest</code>	Destination string.
<code>pszScr</code>	String to append.
<code>ulMaxLen</code>	Maximum number of characters to append.

Return Value -

Example

```
UCHAR sz1[80];

_strcpy (sz1, "123");
_strncat (sz1, "456", 1000);
_MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);

// result: "123456"

_strcpy (sz1, "123");
_strncat (sz1, "456", 2);
_MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);

// result: "12345"
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 147 of: 187
Project: GSEOS		

13.11 `_strncmp ()`

Prototype extern LONG `_strncmp` (CONST UCHAR &psz1[],
 CONST UCHAR &psz2[],
 ULONG ulMaxLen);

Description Compares one string to another.
The function performs an unsigned comparison of psz1 to psz2, starting with the first character in each string and continuing with subsequent characters until the corresponding characters differ or until the end of the strings or ulMaxLen characters are reached.

Parameters

psz1	String 1.
psz2	String 2.
ulMaxLen	Maximum number of characters to compare.

Return Value

< 0	If psz1 is less than psz2.
= 0	If psz1 the same as psz2.
> 0	If psz1 greater than psz2.

Example

```

LONG l;

l = _strncmp ("abcX", "abcY", 3);
if (l == 0)
    _Beep ();

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 148 of: 187
Project: GSEOS		

13.12 `_strncmpi ()`

Prototype extern LONG `_strncmpi` (CONST UCHAR &psz1[],
 CONST UCHAR &psz2[],
 ULONG ulMaxLen);

Description Compares one string to another, without case sensitivity.
 The function performs an unsigned comparison of psz1 to psz2, without case sensitivity, starting with the first character in each string and continuing with subsequent characters until the corresponding characters differ or until the end of the strings or ulMaxLen characters are reached.

Parameters

psz1	String 1.
psz2	String 2.

Return Value

< 0	If psz1 is less than psz2.
= 0	If psz1 the same as psz2.
> 0	If psz1 greater than psz2.

Example

```

LONG l;

l = _strncmpi ("AbcX", "abcY", 3);
if (l == 0)
  _Beep ();
  
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 149 of: 187
Project: GSEOS		

13.13 `_strncpy ()`

Prototype extern VOID `_strncpy` (UCHAR &pszDest[],
 CONST UCHAR &pszScr[],
 ULONG ulMaxLen);

Description Copies one string into another.
 The function copies at most, the first `ulMaxLen` characters of `pszScr` to `pszDest`. If a null character appears in `pszScr` before `ulMaxLen` characters are appended, the function copies all characters from `pszScr`, including the terminating null character. The resulting string is terminated with a null character.
 Overflow checking is performed when strings are copied or appended. The behaviour of the function is undefined if the source and destination strings overlap.
Note: Different behavior than the corresponding C-Function! In the original C-Function the target string `pszDest` might not be null-terminated if the length of `pszScr` is `ulMaxLen` or more.

Parameters

<code>pszDest</code>	Destination string.
<code>pszScr</code>	String to copy.
<code>ulMaxLen</code>	Maximum number of characters to append.

Return Value -

Example

```

UCHAR sz1[80];

_strncpy (sz1, "123", 1000);
_MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);

// result: "123"

_strncpy (sz1, "123", 2);
_MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);

// result: "12"
  
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 150 of: 187
Project: GSEOS		

13.14 `_strnset ()`

Prototype extern VOID `_strnset` (UCHAR &psz[],
 UCHAR uc,
 ULONG ulMaxLen);

Description Set characters of a string to a character.
 The function sets, at most, the first ulMaxLen characters of string psz to uc, except the terminating null character. If ulMaxLen is greater than the length of string, the length of string is used instead of ulMaxLen.

Parameters

psz	String to convert.
uc	Character setting.
ulMaxLen	Maximum number of characters to set.

Return Value -

Example

```

UCHAR sz1[80];

_strcpy (sz1, "12345");
_strnset (sz1, 'x', 1000);
_MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);

// result: "xxxxxx"

_strcpy (sz1, "12345");
_strnset (sz1, 'x', 3);
_MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);

// result: "xxx45"

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 151 of: 187
Project: GSEOS		

13.15 `_strrev ()`

Prototype extern VOID `_strrev (UCHAR &psz[]);`

Description Reverse characters of a string.
The function reverses the order of the characters in string psz. The terminating null character remains in place.

Parameters psz String to convert.

Return Value -

Example UCHAR sz1[80];

 `_strcpy (sz1, "123");`
 `_strrev (sz1);`
 `_MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);`

 // result: "321"

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 152 of: 187
Project: GSEOS		

13.16 `_strset ()`

Prototype extern VOID `_strset` (UCHAR &psz[],
 UCHAR uc);

Description Set characters of a string to a character.
The function sets all the characters of string psz to uc, except the terminating null character.

Parameters psz String to convert.
 uc Character setting.

Return Value -

Example UCHAR sz1[80];

 _strcpy (sz1, "12345");
 _strset (sz1, 'x');
 _MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);

 // result: "xxxxxx"

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 153 of: 187
Project: GSEOS		

13.17 `_strupr ()`

Prototype `extern VOID _strupr (UCHAR &psz[]);`

Description Converts lowercase letters in a string to uppercase.
 The function converts lowercase letters (a to z) to uppercase letters (A to Z). The modification is done in place. No other characters are changed.

Parameters `psz` String to convert.

Return Value -

Example `UCHAR sz1[80];`
 `_strcpy (sz1, "AbCdE123");`
 `_strupr (sz1);`
 `_MessageBox ("Test", sz1, MB_OK | MB_ICONINFORMATION);`
 `// result: "ABCDE123"`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1 style="margin: 0;">GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 154 of: 187
Project: GSEOS		

14 64-Bit Integer Functions

The GSEOS provides functions for 64-bit integer arithmetic. Functions like `_FileSeek ()` or `_GetSystemTimei64 ()` uses 64-bit integers.

Because the GSEOS built in G-Language can not handle 64-bit integers directly, the 64-bit values are stored in a struct. There are two structs to represent 64-bit values.

- For signed 64-bit integers:

```
typedef struct
{
    ULONG ulLow;
    LONG lHigh;
} tsINT64;
```

- For unsigned 64-bit integers:

```
typedef struct
{
    ULONG ulLow;
    ULONG ulHigh;
} tsUINT64;
```

To store constants the two parts of the struct have to be set separately:

```
tsINT64 i64;

// set i64 to -42
i64.ulLow = -42;
i64.lHigh = -1;
```

Furthermore converting functions (see section 11 Converting Functions) may be used to transform 64-bit integer into other formats (`_atoi64 ()`, `_dtoi64 ()`, `_i64toa ()`, `_i64tod ()`, etc.)

IDA Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004
		Issue: 1.12 Date: 5/25/2004
Project: GSEOS		Page: 155 of: 187

14.1 `_i64Add ()`

Prototype extern VOID `_i64Add (CONST tsINT64 &i64A,
 CONST tsINT64 &i64B,
 tsINT64 &i64Result);`

Description The function adds two signed 64-bit integers.
`i64Result = i64A + i64B`
The result is stored in a 64-bit signed integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

<code>i64A</code>	<code>The first parameter.</code>
<code>i64B</code>	<code>The second parameter.</code>
<code>i64Result</code>	<code>The result of the calculation.</code>

Return Value -

Example

```

tsINT64 i64A, i64B, i64Result;
UCHAR   szBuffer[128];

i64A.ulLow = 42;
i64A.lHigh = 0;
i64B.ulLow = -10;
i64B.lHigh = -1;

_i64Add (i64A, i64B, i64Result);
_i64toa (i64Result, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: 32

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h2>GSEOS Language Runtime Library Description</h2>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004 Page: 156 of: 187

14.2 _i64Cmp ()

Prototype extern LONG _i64Cmp (CONST tsINT64 &i64A,
 CONST tsINT64 &i64B);

Description The function compares two signed 64-bit integers and returns a value that is:

Parameters i64A The first parameter.

 i64B The second parameter.

Return Value -1 If i64A is less than i64B

 0 If i64A is the same as i64B

 1 If i64A is greater than i64B

Example

```
tsINT64 i64A, i64B;
UCHAR szBuffer[128];

i64A.ulLow = 42;
i64A.lHigh = 0;
i64B.ulLow = 41;
i64B.lHigh = 0;

if (_i64Cmp (i64A, i64B) > 0)
    _Beep ();
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 157 of: 187
Project: GSEOS		

14.3 `_i64Div ()`

Prototype extern VOID `_i64Div` (CONST tsINT64 &i64A,
 CONST tsINT64 &i64B,
 tsINT64 &i64Result);

Description The function divides two signed 64-bit integers.
`i64Result = i64A / i64B`
 The result is stored in a 64-bit signed integer. It is possible to use the same 64-bit integer struct for both input and result.
 If `i64B` is equal zero an error will be logged and the result is set to zero.

Parameters

<code>i64A</code>	The first parameter.
<code>i64B</code>	The second parameter.
<code>i64Result</code>	The result of the calculation.

Return Value -

Example

```

tsINT64 i64A, i64B, i64Result;
UCHAR   szBuffer[128];

i64A.ulLow = 42;
i64A.lHigh = 0;
i64B.ulLow = 2;
i64B.lHigh = 0;

_i64Div (i64A, i64B, i64Result);
_i64toa (i64Result, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: 21

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 158 of: 187
Project: GSEOS		

14.4 `_i64Mul ()`

Prototype extern VOID `_i64Mul` (CONST tsINT64 &i64A,
 CONST tsINT64 &i64B,
 tsINT64 &i64Result);

Description The function multiplies two signed 64-bit integers.
`i64Result = i64A * i64B`

The result is stored in a 64-bit signed integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

<code>i64A</code>		The first parameter.
<code>i64B</code>		The second parameter.
<code>i64Result</code>		The result of the calculation.

Return Value -

Example

```

tsINT64 i64A, i64B, i64Result;
UCHAR  szBuffer[128];

i64A.ulLow = 42;
i64A.lHigh = 0;
i64B.ulLow = -2;
i64B.lHigh = -1;

_i64Mul (i64A, i64B, i64Result);
_i64toa (i64Result, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: -84

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004
		Issue: 1.12 Date: 5/25/2004
Project: GSEOS		Page: 159 of: 187

14.5 _i64Shra ()

Prototype extern VOID _i64Shra (CONST tsINT64 &i64,
 tsINT64 &i64Result,
 ULONG ulShiftCount);

Description The function performs a right arithmetic shift operation on a signed 64-bit integer.

i64Result = i64A >> ulShiftCount

The shift count is the number of bit positions that the value's bits move.

In a right arithmetic shift operation on a signed value, the value's bits move to the right, and vacated bits on the left side of the value are set to the value of the sign bit.

The result is stored in a 64-bit signed integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

i64	The value to shift.
i64Result	The result of the calculation.
ulShiftCount	The number of bit positions that the value's bits move.

Return Value -

Example

```
tsINT64 i64A, i64Result;
UCHAR  szBuffer[128];

i64A.ulLow = -42;
i64A.lHigh = -1;

_i64Shra (i64A, i64Result, 1);
_i64toa (i64Result, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: -21
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 160 of: 187
Project: GSEOS		

14.6 `_i64Sub ()`

Prototype `extern VOID _i64Sub (CONST tsINT64 &i64A,
 CONST tsINT64 &i64B,
 tsINT64 &i64Result);`

Description The function subtracts two signed 64-bit integers.
`i64Result = i64A - i64B`
The result is stored in a 64-bit signed integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

<code>i64A</code>	The first parameter.
<code>i64B</code>	The second parameter.
<code>i64Result</code>	The result of the calculation.

Return Value -

Example

```
tsINT64 i64A, i64B, i64Result;
UCHAR   szBuffer[128];

i64A.ulLow = 42;
i64A.lHigh = 0;
i64B.ulLow = 50;
i64B.lHigh = 0;

_i64Sub (i64A, i64B, i64Result);
_i64toa (i64Result, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: -8
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1 style="margin: 0;">GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 162 of: 187
Project: GSEOS		

14.8 `_ui64And ()`

Prototype extern VOID `_ui64And` (CONST tsUINT64 &ui64A,
 CONST tsUINT64 &ui64B,
 tsUINT64 &ui64Result);

Description The function provides a bitwise AND for two unsigned 64-bit integers.
`ui64Result = ui64A & ui64B`
The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

<code>ui64A</code>	<code>The first parameter.</code>
<code>ui64B</code>	<code>The second parameter.</code>
<code>ui64Result</code>	<code>The result of the calculation.</code>

Return Value -

Example

```
tsUINT64 ui64A, ui64B, ui64Result;
UCHAR    szBuffer[128];

ui64A.ulLow  = 0x76543210;
ui64A.ulHigh = 0xFEDCBA98;
ui64B.ulLow  = 0xF0F0F0F0;
ui64B.ulHigh = 0xF0F0F0F0;

_ui64And (ui64A, ui64B, ui64Result);
_ui64toa (ui64Result, szBuffer, 16);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: F0D0B09070503010
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 165 of: 187

14.11 `_ui64Mul ()`

Prototype extern VOID `_ui64Mul` (CONST tsUINT64 &ui64A,
 CONST tsUINT64 &ui64B,
 tsUINT64 &ui64Result);

Description The function multiplies two unsigned 64-bit integers.

`ui64Result = ui64A * ui64B`

The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

<code>ui64A</code>	The first parameter.
<code>ui64B</code>	The second parameter.
<code>ui64Result</code>	The result of the calculation.

Return Value -

Example

```
tsUINT64 ui64A, ui64B, ui64Result;  
UCHAR    szBuffer[128];  
  
ui64A.ulLow  = 42;  
ui64A.ulHigh = 0;  
ui64B.ulLow  = 2;  
ui64B.ulHigh = 0;  
  
_ui64Mul (ui64A, ui64B, ui64Result);  
_ui64toa (ui64Result, szBuffer, 10);  
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);  
  
// result: 84
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 166 of: 187
Project: GSEOS		

14.12 `_ui64Not ()`

Prototype extern VOID `_ui64Not (CONST tsUINT64 &ui64A,
 tsUINT64 &ui64Result);`

Description The function provides a bitwise NOT for an unsigned 64-bit integers.
`ui64Result = ~ui64A`
The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters ui64A The first parameter.

 ui64Result The result of the calculation.

Return Value -

Example

```
tsUINT64 ui64A, ui64Result;
UCHAR    szBuffer[128];

ui64A.ulLow  = 0x76543210;
ui64A.ulHigh = 0xFEDCBA98;

_ui64Not (ui64A, ui64Result);
_ui64toa (ui64Result, szBuffer, 16);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: 0123456789ABCDEF
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 167 of: 187
Project: GSEOS		

14.13 `_ui64Or ()`

Prototype extern VOID `_ui64Or` (CONST tsUINT64 &ui64A,
 CONST tsUINT64 &ui64B,
 tsUINT64 &ui64Result);

Description The function provides a bitwise OR for two unsigned 64-bit integers.
`ui64Result = ui64A | ui64B`
The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

<code>ui64A</code>	The first parameter.
<code>ui64B</code>	The second parameter.
<code>ui64Result</code>	The result of the calculation.

Return Value -

Example

```
tsUINT64 ui64A, ui64B, ui64Result;
UCHAR    szBuffer[128];

ui64A.ulLow  = 0x76543210;
ui64A.ulHigh = 0xFEDCBA98;
ui64B.ulLow  = 0xF0F0F0F0;
ui64B.ulHigh = 0xF0F0F0F0;

_ui64Or (ui64A, ui64B, ui64Result);
_ui64toa (ui64Result, szBuffer, 16);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: FEFCFAF8F6F4F2F0
```

14.14 _ui64Rol ()

Prototype extern VOID _ui64Rol (CONST tsUINT64 &ui64,
tsUINT64 &ui64Result,
ULONG ulRotateCount);

Description The function performs a left rotate operation on an unsigned 64-bit integer. The left rotate operation on an unsigned value moves the value's bits to the left, except for the top bit, which is returned to the right. The rotate is repeated the number of times indicated by ulRotateCount. The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

ui64	The value to shift.
ui64Result	The result of the calculation.
ulRotateCount	The number of bit positions that the value's bits rotate.

Return Value -

Example

```
tsUINT64 ui64A, ui64Result;  
UCHAR    szBuffer[128];  
  
ui64A.ulLow  = 0x76543210;  
ui64A.ulHigh = 0xFEDCBA98;  
  
_ui64Rol (ui64A, ui64Result, 4);  
_ui64toa (ui64Result, szBuffer, 16);  
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);  
  
// result: 0xEDCBA9876543210F
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 169 of: 187
Project: GSEOS		

14.15 `_ui64Ror ()`

Prototype extern VOID `_ui64Ror` (CONST tsUINT64 &ui64,
 tsUINT64 &ui64Result,
 ULONG ulRotateCount);

Description The function performs a right rotate operation on an unsigned 64-bit integer. The right rotate operation on an unsigned value moves the value's bits to the right, expect for the bottom bit, which is returned to the left. The rotate is repeated the number of times indicated by `ulRotateCount`. The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters `ui64` The value to shift.
`ui64Result` The result of the calculation.
`ulRotateCount` The number of bit positions that the value's bits rotate.

Return Value -


Example

```
tsUINT64 ui64A, ui64Result;
UCHAR    szBuffer[128];

ui64A.ulLow  = 0x76543210;
ui64A.ulHigh = 0xFEDCBA98;

_ui64Ror (ui64A, ui64Result, 4);
_ui64toa (ui64Result, szBuffer, 16);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: 0x0FEDCBA987654321
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<p style="text-align: center;">GSEOS Language Runtime Library Description</p>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 170 of: 187

14.16 `_ui64Shl ()`

Prototype `extern VOID _ui64Shl (CONST tsUINT64 &ui64,
 tsUINT64 &ui64Result,
 ULONG ulShiftCount);`

Description The function performs a left logical shift operation on an unsigned 64-bit integer.
`ui64Result = ui64A << ulShiftCount`
The shift count is the number of bit positions that the value's bits move.
In a left logical shift operation on an unsigned value, the value's bits move to the left, and vacated bits on the right side of the value are set to zero.
The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

<code>ui64</code>	The value to shift.
<code>ui64Result</code>	The result of the calculation.
<code>ulShiftCount</code>	The number of bit positions that the value's bits move.

Return Value -

Example

```
tsUINT64 ui64A, ui64Result;
UCHAR    szBuffer[128];

ui64A.ulLow  = 42;
ui64A.ulHigh = 0;

_ui64Shl (ui64A, ui64Result, 1);
_ui64toa (ui64Result, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: 84
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 171 of: 187
Project: GSEOS		

14.17 `_ui64Shrl ()`

Prototype extern VOID `_ui64Shrl (CONST tsUINT64 &ui64,
 tsUINT64 &ui64Result,
 ULONG ulShiftCount);`

Description The function performs a right logical shift operation on an unsigned 64-bit integer.
`ui64Result = ui64A >> ulShiftCount`
 The shift count is the number of bit positions that the value's bits move.
 In a right logical shift operation on an unsigned value, the value's bits move to the right, and vacated bits on the left side of the value are set to zero.
 The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

<code>ui64</code>	The value to shift.
<code>ui64Result</code>	The result of the calculation.
<code>ulShiftCount</code>	The number of bit positions that the value's bits move.

Return Value -

Example

```
tsUINT64 ui64A, ui64Result;
UCHAR    szBuffer[128];

ui64A.ulLow  = 42;
ui64A.ulHigh = 0;

_ui64Shrl (ui64A, ui64Result, 1);
_ui64toa (ui64Result, szBuffer, 10);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: 21
```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 173 of: 187
Project: GSEOS		

14.19 `_ui64Xor ()`

Prototype extern VOID `_ui64Xor` (CONST tsUINT64 &ui64A,
 CONST tsUINT64 &ui64B,
 tsUINT64 &ui64Result);

Description The function provides a bitwise XOR for two unsigned 64-bit integers.
`ui64Result = ui64A ^ ui64B`
The result is stored in a 64-bit unsigned integer. It is possible to use the same 64-bit integer struct for both input and result.

Parameters

<code>ui64A</code>	The first parameter.
<code>ui64B</code>	The second parameter.
<code>ui64Result</code>	The result of the calculation.

Return Value -

Example

```
tsUINT64 ui64A, ui64B, ui64Result;
UCHAR    szBuffer[128];

ui64A.ulLow  = 0x76543210;
ui64A.ulHigh = 0xFEDCBA98;
ui64B.ulLow  = 0xF0F0F0F0;
ui64B.ulHigh = 0xF0F0F0F0;

_ui64Xor (ui64A, ui64B, ui64Result);
_ui64toa (ui64Result, szBuffer, 16);
_MessageBox ("Test", szBuffer, MB_OK | MB_ICONINFORMATION);

// result: E2C4A6886A4C2E0
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 174 of: 187

15 Multimedia Functions

Microsoft Windows Media Control Interface (MCI) provides standard commands for playing multimedia devices and recording multimedia resource files. These commands are a generic interface to nearly every kind of multimedia device. GSEOS can use MCI to control any supported multimedia device, including waveform-audio devices, MIDI sequencers, CD audio devices, and digital-video (video playback) devices.

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 175 of: 187
Project: GSEOS		

15.1 _MciPlay ()

Prototype extern BOOL _MciPlay (CONST UCHAR &pszFileName[]);

Description The function starts playing a media file.

The playing device is based on the extension used by the file. The media file is played asynchronously and the function returns immediately.

The function is an easy-to-use predefined sequence of MCI commands. To provide the full playback functionality (pause, stop playing, repeat, reverse, play CD-Audio Track, etc.) you should use the function _MciSendString() (see section 15.2).

Parameters pszFileName A null-terminated string that specifies the media file name.

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If the function fails, the return value is zero.

Example

```
// lets user choose a media file to play
UCHAR szBuffer[1024];

if (_DlgGetOpenFileName (szBuffer, sizeof (szBuffer)))
    _MciPlay (szBuffer);

// play two videos simultaneously
_MciPlay ("\\data\\media\\MyVideo1.avi");
_MciPlay ("c:\\MyMovies\\MyVideo2.mpg");
```

GSEOS Language Runtime Library Description

15.2 _MciSendString ()

Prototype extern BOOL _MciSendString (CONST UCHAR &pszCommandString[],
 UCHAR &pszReturnString[]);

Description The function sends a command string to an MCI device. The device that the command is sent to is specified in the command string.

MCI command strings use a consistent verb-object-modifier syntax. Each command string includes a command, a device identifier, and command arguments. Arguments are optional for some commands and required for others.

For more information about the command strings, see Microsoft Windows SDK documentation.

Parameters pszCommandString A null-terminated string that specifies a MCI command string.

 pszReturnString A buffer that receives return information.

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If the function fails, the return value is zero.

Example

```

// play a wave file
UCHAR aucReturnString[1024];

_MciSendString ("open data\\media\\beep.wav alias MySound wait",
                aucReturnString);
_MciSendString ("play MySound wait", aucReturnString);
_MciSendString ("close MySound wait", aucReturnString);

// close all media streams currently in use (stop all playing)
_MciSendString ("close all", aucReturnString);

```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 177 of: 187

16 Miscellaneous Functions

This section describes the various functions not fitting in the other chapter.

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 178 of: 187
Project: GSEOS		

16.1 _AddMessage ()

Prototype extern VOID _AddMessage (ULONG ulCategory,
 CONST UCHAR &pszSource[],
 CONST UCHAR &pszMessage[]);

Description The function adds a message to the log file.

Parameters

ulCategory	Specifies the category of the message:
	MSG_INFORMATION Information
	MSG_WARNING Warning
	MSG_ERROR Error
pszSource	Points to a null-terminated string containing the source description of the message.
pszMessage	Points to a null-terminated string containing the message text.

Return Value -

Example

```

VOID TestDlgAddMessage ()
{
    UCHAR aucBuffer[1024];
    ULONG ulReturn;

    if (_DlgGetString ("Enter Text.", aucBuffer, sizeof (aucBuffer),
        ulReturn))
    {
        _AddMessage (MSG_INFORMATION, "Test", aucBuffer);
    }
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 179 of: 187

16.2 _Beep ()

Prototype `extern VOID _Beep ();`

Description The function produces a simple notification sound.

Parameters -

Return Value -

Example

```
int i;
for (i = 0; i < 3; ++i)
    _Beep ();
```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 180 of: 187
Project: GSEOS		

16.3 _GetBdmQueueFillState ()

Prototype extern ULONG _GetBdmQueueFillState ();

Description The function returns the current BDM queue fill state in bytes.

Parameters -

Return Value Current BDM queue fill state in bytes.

Example

```

decode on (_Timer1Sec)
{
    if (_GetBdmQueueFillState () > 1024 * 1024)
        DppSimReadEnable (FALSE);
    else
        DppSimReadEnable (TRUE);
}

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 181 of: 187
Project: GSEOS		

16.4 _GetBlockId ()

- Prototype** `extern ULONG _GetBlockId (CONST UCHAR &pszBlockName[]);`
- Description** The function returns the BDM Block Id for a given block name. For IDA internal use only.
- Parameters** `pszBlockName` BDM Block name
- Return Value** BDM internal block number. On invalid Block name the function returns 0xFFFFFFFF.
- Example** `ULONG ulId;`
`ulId = _GetBlockId ("MyBlockName");`
`if (ulId != 0xFFFFFFFF)`
`SetBlockInMyExternalDll (ulId, sizeof ("MyBlockName"));`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	<h1>GSEOS Language Runtime Library Description</h1>	Ref.: IDA-GSEOS-0004
Project: GSEOS		Issue: 1.12 Date: 5/25/2004
		Page: 182 of: 187

16.5 _GetQueueDriverHandle ()

Prototype `extern ULONG _GetQueueDriverHandle ();`

Description The function returns the current Queue Driver Handle. For IDA internal use only.

Parameters -

Return Value Current Queue Driver Handle.

Example `ULONG ulHandle;`

`ulHandle = _GetQueueDriverHandle ();`
`MyVerySpecialDllFunction (ulHandle, 42);`

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 184 of: 187
Project: GSEOS		

MB_ICONQUESTION A question-mark icon appears in the message box.

MB_ICONERROR A stop-sign icon appears in the message box.

Return Value If the function succeeds, the return value is one of the following values returned by the dialog box:

- | | |
|----------|-----------------------------|
| IDOK | OK button was selected. |
| IDCANCEL | Cancel button was selected. |
| IDABORT | Abort button was selected. |
| IDRETRY | Retry button was selected. |
| IDIGNORE | Ignore button was selected. |
| IDYES | Yes button was selected. |
| IDNO | No button was selected. |

Example

```

VOID TestMessageBox ()
{
  if (_MessageBox ("Test", "Beep or not Beep?",
                  MB_YESNO | MB_ICONQUESTION) == IDYES)
  {
    _MessageBox ("Test", "Ok, let's Beep!", MB_OK | MB_ICONERROR);
    _Beep ();
  }
}

```


 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 185 of: 187
Project: GSEOS		

16.7 _StartApplication ()

Prototype extern BOOL _StartApplication (CONST UCHAR &pszCommandLine[]);

Description The function is used to run a new program.

Parameters pszCommandLine A null-terminated string that specifies the command line to execute. The first white space-delimited token of the command line specifies the application name. The current directory of the new application is set to directory of GSEOS.EXE.

Return Value TRUE If the function succeeds, the return value is nonzero.

 FALSE If the function fails, the return value is zero.

Example

```

_StartApplication ("notepad");
_StartApplication ("notepad.exe c:\\temp\\test.txt");
_StartApplication ("ocl data\\test.ocl");
_StartApplication ("c:\\program files\\idl\\idl.exe");

// displays directory, console remains open
_StartApplication ("cmd /k dir");

// displays directory, console terminates
_StartApplication ("cmd /c dir");

```

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 186 of: 187
Project: GSEOS		

16.8 _SetThreadPriority ()

Prototype extern BOOL _SetThreadPriority (LONG lPriority);

Description The function sets the priority value for the current thread.

The operating system uses the base priority level of all executable threads to determine which thread gets the next slice of CPU time. Threads are scheduled in a round-robin fashion at each priority level, and only when there are no executable threads at a higher level does scheduling of threads at a lower level take place.

The _SetThreadPriority() function enables setting the base priority level of a thread relative to the priority class of its process. The priority class of the GSEOS process is set by GSEOS/Control Panel/BDM/Settings/Program Priority.

Note: The function will change the priority level for GSEOS batch files only. You can't change the priority neither decoder functions nor screen elements. Use with care. A too high priority of a batch file may degrade overall system performance.

Parameters lPriority Priority value for the thread. This parameter can be one of the following values:

```

THREAD_PRIORITY_IDLE
THREAD_PRIORITY_LOWEST
THREAD_PRIORITY_BELOW_NORMAL (default for batches)
THREAD_PRIORITY_NORMAL
THREAD_PRIORITY_ABOVE_NORMAL
THREAD_PRIORITY_HIGHEST
THREAD_PRIORITY_TIME_CRITICAL

```

Return Value TRUE If the function succeeds, the return value is nonzero.

FALSE If the function fails, the return value is zero.

Example // in a batch file
// set the highest available priority

_SetThreadPriority (THREAD_PRIORITY_TIME_CRITICAL);

 Institut für Datentechnik und Kommunikationsnetze TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG	GSEOS Language Runtime Library Description	Ref.: IDA-GSEOS-0004 Issue: 1.12 Date: 5/25/2004 Page: 187 of: 187
Project: GSEOS		

16.9 _SyncIdx ()

Prototype extern BOOL _SyncIdx (CONST UCHAR &ucData[],
 CONST UCHAR &ucSync[],
 ULONG &pulIdx);

Description The function searches a sync pattern in a byte stream.

Parameters

ucData	Points to data stream (array of characters).
ucSync	Points to sync pattern (array of characters).
pulIdx	Points to the return buffer. On success the buffer is filled with the index of the beginning of the first sync pattern found in the input data stream.

Return Value TRUE If the function succeeds, the return value is nonzero.
 FALSE If the function fails, the return value is zero.

Example

```
// search sync
if (_SyncIdx (aucSyncBuffer[0, ulSyncBufferIdx],
             aucSync[0, ulSyncSize], ulSyncIdx))
{
  // sync found, copy to destination block
  ulToCopy = ulSyncBufferIdx - ulSyncIdx;
  blkDestination.auc[0, ulToCopy] =
    aucSyncBuffer[ulSyncIdx, ulToCopy];
  ulDestinationIdx = ulToCopy;
  ulState = 2;
}
else
{
  // no sync found, discard data
  if (ulSyncBufferIdx >= ulSyncSize)
  {
    ulToCopy = ulSyncSize - 1;
    aucSyncBuffer[0, ulToCopy] =
      aucSyncBuffer[ulSyncBufferIdx - ulToCopy, ulToCopy];
    ulSyncBufferIdx = ulToCopy;
  }
}
```