

EDIABAS - API FUNCTION PRIMER

EDIABAS

Electronic Diagnostic Basic System

API FUNCTION PRIMER

VERSION 6c

Copyright BMW AG, created by Softing AG

APIREF.DOC

EDIABAS - API FUNCTION PRIMER

CONTENTS

CONTENTS	2
1. Update history	4
2. Introduction	5
2.1. About this Manual	5
2.2. Conventions	5
2.3. Special features, definitions, acronyms	6
3. Data types, constants and error messages	7
3.1. Data types	7
3.2. Constants	8
3.3. Error messages	9
4. API function Primer	11
4.1. Using the function primer	11
4.2. Functions	11
apiBreak	12
apiCallBack	13
apiEnd	14
apiErrorCode	15
apiErrorHandler	16
apiErrorText	17
apiGetConfig	18
apiInit	19
apiJob	22
apiJobData	23
apiJobExt	24
apiJobInfo	25
apiResultBinary	26
apiResultByte	27

EDIABAS - API FUNCTION PRIMER

- apiResultChar 28
- apiResultDWord 29
- apiResultFormat 30
- apiResultInt 31
- apiResultLong 32
- apiResultName 33
- apiResultNumber 34
- apiResultReal 35
- apiResultsDelete 36
- apiResultSets 37
- apiResultsNew 38
- apiResultsScope 39
- apiResultText 40
- apiResultVar 41
- apiResultWord 42
- apiSetConfig 43
- apiState 44
- apiSwitchDevice 45
- 5. Other Functions 46**
 - enableServer 46
 - closeServer 47
- A. List of references 48**

EDIABAS - API FUNCTION PRIMER

1. Update history

Version 3.0 First release

Version 4.1 Revised for EDIABAS V4.1.0

Version 5 Inclusion of the new API functions apiJobExt and apiJobInfo

Version 5a API functions apiResultByte/apiResultChar under VB 4.0 available

Version 6 New API function apiInitExt as supplement to apiJobInfo

Version 6c Revised for EDIABAS V6.4.4

2. Introduction

2.1. About this Manual

This manual describes the constants and functions that are available in the EDIABAS-API interface. Reference [5] gives a detailed description of their use. Linking into a development environment is described in Reference [6]. You will find general information about EDIABAS and control unit description files in Reference [4].

2.2. Conventions

The following typographical conventions are used in this manual:

Example	Description
SAMPLE.C	Upper case characters are used for filenames, registers and operating system commands.
apiJob, APIREADY	Bold type is used for key words and operators of the BEST/2 and BEST/1 languages and for API functions. In syntax descriptions these words must be written as shown.
<i>expression</i>	Italics designate placeholders for values to be entered by the programmer; e.g., file names.
[option]	Words enclosed in square brackets may be optionally specified.
{ result argument }	Curvy braces and vertical strokes characterize entries from which only one must be selected, except when in square brackets.
[constant...] job...	An ellipsis (three dots) which directly follows an expression indicates that several expressions of the same type can follow.
hallo="Test";	This syntax designates examples, user entries, program outputs and error messages.

EDIABAS - API FUNCTION PRIMER

<code>while() {</code>	A column or a row comprising three dots
<code>.</code>	indicates that a section of an example was
<code>.}</code>	intentionally omitted.
<code>[1]</code>	Reference to a document in References.

2.3. Special features, definitions, acronyms

The abbreviations used in this and all other EDIABAS documents are explained in the "GLOSSARY" section of the "EDIABAS User Manual".

3. Data types, constants and error messages

3.1. Data types

API provides the following predefined data types.

Data type	Definition (Language C/C++)	Definition (Visual Basic)	Explanation
APIBOOL	enum , values: APIFALSE, APITRUE	Integer	Return value
APICHAR	char	-	Result type
APIBYTE	unsigned char	-	Result type
APIINTEGER	short	Integer	Result type
APIWORD	unsigned short	Integer	Result type
APILONG	long	Long	Result type
APIDWORD	unsigned long	Long	Result type
APIREAL	double	Double	Result type
APITEXT	APICHAR	String	Result type
APIBINARY	APIBYTE	String	Result type

As well as the simple data types, this primer also defines the complex data type **APIRESULTFIELD** which represents a reference to a result field.

Variables of the **APIRESULTFIELD** data type can be used like variables of simple data types (e.g. **APIBYTE**).

All API data types are not available under Visual Basic.

3.2. Constants

API provides the following defined constants:

Constant	Purpose
APIMAXCONFIG	Maximum recommended buffer size for <i>configurationsetting</i> (including termination character)
APIMAXDEVICE	Maximum length of device arguments <i>connection</i> and <i>application</i> (including last character)
APIMAXNAME	Maximum length of job arguments <i>control unit</i> and <i>name</i> , result names and format strings (including last character)
APIMAXSTDPARA	Maximum length of the job argument <i>Parameter</i> for standard jobs
APIMAXPARA	Maximum length of job argument <i>parameter</i> (including last character if argument exists as a string)
APIMAXRESULT	Maximum length of job argument <i>result</i> (including last character)
APIMAXTEXT	Maximum and recommended size of <i>APITEXT</i> result buffer (including last character)
APIMAXBINARY	Maximum and recommended size of <i>APIBINARY</i> result buffer
APIMAXFILENAME	Maximum and recommended size of file names (including final character)
APIBUSY Visual Basic: APIBUSY_	BUSY status of EDIABAS
APIREADY Visual Basic: APIREADY_	READY status of EDIABAS
APIBREAK Visual Basic: APIBREAK_	BREAK status of EDIABAS

EDIABAS - API FUNCTION PRIMER

APIERROR ERROR status of EDIABAS

Visual Basic: **APIERROR_**

EDIABAS_XXX_#### Error messages (see section "Error messages")

3.3. Error messages

An EDIABAS error message comprises the following elements:

Error symbol (error origin, error number)

Error code

Default error text

When an error occurs the application program can get the error code from EDIABAS with the function **apiErrorCode()** and/or the default error text with the function **apiErrorText()**.

Each error code is assigned an error symbol that can be referenced as part of error handling in the application program.

The error symbols are defined in API.H.

The error symbol of each error has the following structure:

EDIABAS_XXX_####

The XXX group of characters denotes the EDIABAS-internal origin area of the error, and the number #### is the corresponding error number.

Error and program areas:

EDIABAS_API_####	API
EDIABAS_BIP_####	BEST interpreter, run-time system
EDIABAS_SYS_####	Sequential control, run-time system
EDIABAS_IFH_####	Interface handler
EDIABAS_NET_####	Network, run-time system
EDIABAS_RUN_####	ECU description file

EDIABAS - API FUNCTION PRIMER

The pseudo-error **EDIABAS_ERR_NONE** (error code 0) is an exception, it indicates an error-free condition.

Document [1] gives a full list of errors together with their error symbol, error code, default error text and a description of the error message.

4. API function Primer

4.1. Using the function primer

Each function description consists of four sections:

- **Summary** A brief description of the function, giving syntax and arguments.
- **Remarks** A detailed description of the function and its use.
- **Return** Describes the value returned by the function.
- **Interface** Lists all interfaces which support the functions (C/C++, Visual Basic and DLL interface). A detailed description on these interfaces can be found in reference [6].
- **See also** Related functions.

4.2. Functions

The API functions are described in alphabetical order in this section.

apiBreak

Summary Aborts job in progress

void apiBreak (void)

Remarks The function **apiBreak()** can be used to abort an ongoing job execution (automatic call with **apiEnd()**). Any results are lost.

Return -

Interface C/C++ Visual Basic DLL

See also **apiEnd, apiInit, apiInitExt, apiState, apiCallBack**

apiCallBack

Summary Installs a call-back routine in the application program (optional)

void apiCallBack (APIBOOL (*action)(void))

action function reference (address) of the call-back routine

Remarks The ongoing job execution can be controlled by the **APIBOOL** return of the call-back routine:

Return = **APITRUE** Job is aborted

Return = **APIFALSE** Job is continued

EDIABAS only calls the call-back routine with the functions **apiState** or **apiResultxxx!**

Return -

Interface C/C++ Visual Basic DLL

See also **apiBreak, apiEnd, apilnit, apilnitExt, apiState, apiSwitchDevice**

apiEnd

Summary Cancels the application lock with EDIABAS

void apiEnd (void)

Remarks The EDIABAS run-time system is released for other applications.
The occupied storage location is returned.

Return -

Interface C/C++ Visual Basic DLL

See also **apiBreak, apilnit, apilnitExt, apiState, apiCallBack,**
apiSwitchDevice

apiErrorCode

Summary Identifies the current error code

int apiErrorCode (void)

Remarks This function may also be used out of **apilnit()** and **apilnitExt(...)**. If **apilnit()** or **apilnitExt(...)** return **APIFALSE** you can get the error code by **apiErrorCode()**

Return Error code or **EDIABAS_ERR_NONE** if there is no error

Interface C/C++ Visual Basic DLL

See also **apiErrorHandler, apiErrorText**

apiErrorHandler

Summary Installs an error handler function in the application program (optional).

void apiErrorHandler (void (action*)(void))**

action reference of error handler (function address)

Remarks The error handler function in the application program is called by EDIABAS whenever an EDIABAS error occurs.

Return -

Interface C/C++ ~~Visual Basic~~ ~~DLL~~

See also **apiErrorCode, apiErrorText**

apiErrorText

Summary Identifies the current error text.

const char *apiErrorText (void)

Remarks This function may also be used out of **apilnit()** and **apilnitExt(...)**. If **apilnit()** or **apilnitExt(...)** return **APIFALSE** you can get the error text by **apiErrorText()**

Return Pointer to error text or **NULL** if there is no error.

Interface C/C++ Visual Basic DLL

See also **apiErrorCode, apiErrorHandler, apiErrorText**

apiGetConfig

Summary **Interrogation of the EDIABAS configuration**

APIBOOL apiGetConfig (const char **config*, char **buf*)

config Configuration element to be interrogated

buf Buffer for configuration setting

Remarks **Determination of the setting for *config*. **APIMAXCONFIG** is the maximum number of copied characters and thus the recommended buffer size for *buf*.**

The error status is not affected.

All configuration elements and their settings are described in reference [4].

Return **APITRUE** Configuration determined

APIFALSE Erroneous configuration inquiry

Interface C/C++ Visual Basic DLL

See also **apiSetConfig**

EDIABAS - API FUNCTION PRIMER

See also **apiBreak apiEnd apiState apiCallBack apiSwitchDevice**

apiJob

Summary Sends a job to EDIABAS

void apiJob (char *ecu, char*job, char*para, char*result)

ecu Name of the CU group or variant description file to be loaded
(max. **APIMAXNAME**)

job Job to be sent, the job name is defined in the addressed SGBD
(max. **APIMAXNAME**)

para Job parameter (max. **APIMAXPARA**) as a string, several
parameters must be separated by a semi-colon. The job
parameters depend on the particular job (*job*).

result Results to be identified (max. **APIMAXRESULT**), several results
must be separated by a semi-colon. A blank string ("") must be
defined to process all results. The results depend on the
particular job.

Remarks Sends a job to EDIABAS without waiting for the job to finish executing.
The argument *ecu* is first processed as the name of a variant
description file. If this file (*ecu.prg*) does not exist it is processed as a
group description file (*ecu.grp*)

Return -

Interface C/C++ Visual Basic DLL

See also **apiJobData, apiJobExt**

apiJobData

Summary Sends a job to EDIABAS

**void apiJobData (char *ecu, char*job, unsigned char*parabuf,
int*paralen, char*result)**

ecu Name of the CU group or variant description file to be loaded
(max. **APIMAXNAME**)

job Job to be sent, the job name is defined in the addressed SGBD
(max. **APIMAXNAME**)

parabuf Job parameter (max. **APIMAXPARA**) as binary data. The job
parameters depend on the particular job

paralen Number of data bytes of the job parameter.

result Results to be identified (max. APIMAXRESULT), several results
must be separated by a semi-colon. A blank string ("") must be
defined to process all results. The results depend on the
particular job.

Remarks See **apiJob**.

Return -

Interface C/C++ Visual Basic DLL

See also **apiJob, apiJobExt**

apiJobExt

Summary Issues a job to EDIABAS

```
void apiJobExt (const char *ecu, const char *job,  
               const unsigned char *stdpara, int stdparalen,  
               const unsigned char *para, int paralen,  
               const char *result, long reserved )
```

<i>ecu</i>	Name of the group or variant description file to be loaded (max. APIMAXNAME)
<i>job</i>	Job to be issue; the job name is determined in the addressed ECU description file (max. APIMAXNAME)
<i>stdpara</i>	Job parameter (max. APIMAXSTDPARA) to the standard jobs Initialisierung, Identifikation, Ende. The parameter data depend on the respective job.
<i>stdparalen</i>	Number of data bytes of the job parameter for the standard jobs
<i>para</i>	Job parameters (max. APIMAXPARA) as binary data. The parameter data depend on the respective job.
<i>paralen</i>	Number of data bytes of the job parameter.
<i>result</i>	Results to be determined (max. APIMAXRESULT); multiple results are to be separated with a semicolon. A blank string ("") is to be specified in order to process all results. The results depend on the respective job.
<i>reserved</i>	0, reserved for extensions.

Remarks See **apiJob**, **apiJobData**

Return -

Interface C/C++ Visual Basic DLL

See also **apiJob**, **apiJobData**

apiJobInfo

Summary Interrogations of the processing degree for a job

int apiJobInfo (char **info Text*)

infoText NULL or address of the buffer variable in the application program

Remarks This function returns the processing degree of a job as a percent. This function must be supported in the description file. If not, the function returns -1 as a result (i.e., the processing degree is undefined).

Range of values: 0 ... 100 [%]

In addition, the function **apiJobInfo** allows for the optional query of an information text concerning the processing degree. This function must be supported in the description file.

APIMAXTEXT is the maximum possible length of the information text and, consequently, the recommended buffer size of the target variable.

If the function parameter value NULL is transferred, no additional information text query is carried through.

Return Processing degree in percent

Interface C/C++ Visual Basic DLL

See also -

apiResultBinary

Summary Identifies an APIBINARY result

APIBOOL apiResultBinary (APIBINARY*buf, APIWORD*buflen, char*result, APIWORD set)

<i>buf</i>	Address of buffer target variable in the application programme
<i>buflen</i>	Address of length target variable in the application programme
<i>result</i>	Result to be polled
<i>set</i>	Result set number

Remarks Identifies an APIBINARY result. Waits for an ongoing job to finish processing.

The desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* in the application program. The number of copied bytes is copied to the target variable *buflen*. **APIMAXBINARY** is the maximum number of copied characters of the APIBINARY type and therefore the recommended buffer size of the buffer target variables. If the result is not in APIBINARY format there is no conversion to APIBINARY format.

Affects error status.

Return	APITRUE	Result exists
	APIFALSE	Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultChar, apiResultByte, apiResultInt, apiResultWord, apiResultLong, apiResultDWord, apiResultReal, apiResultText**

apiResultByte

Summary Identifies an APIBYTE result

APIBOOL apiResultByte (APIBYTE*buf, char*result, APIWORD set)

buf Address of target variable in the application programme

result Result to be polled

set Result set number

Remarks Identifies an **APIBYTE** result. Waits for an ongoing job to finish processing. The desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* of the application program. If the result is not in **APIBYTE** format but type conversion is possible, then it is converted to **APIBYTE** format. Affects error status.

Return **APITRUE** Result exists
APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultChar, apiResultInt, apiResultWord, apiResultLong, apiResultDWord, apiResultReal, apiResultText, apiResultBinary**

apiResultChar

Summary Identifies an APICCHAR result

APIBOOL apiResultChar (APICCHAR*buf, char*result, APIWORD set)

buf Address of target variable in the application programme

result Result to be polled

set Result set number

Remarks Identifies an **APICCHAR** result. Waits for an ongoing job to finish processing. The desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* in the application program. If the result is not in **APICCHAR** format but type conversion is possible, then it is converted to **APICCHAR** format. Affects error status.

Return **APITRUE** Result exists
 APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultByte, apiResultInt, apiResultWord, apiResultLong,**
apiResultDWord, apiResultReal, apiResultText, apiResultBinary

apiResultDWord

Summary Identifies an APIDWORD result

APIBOOL apiResultDWord (APIDWORD*buf, char*result, APIWORD set)

buf Address of target variable in the application programme
result Result to be polled
set Result set number

Remarks Identifies an **APIDWORD** result. Waits for an ongoing job to finish processing. The desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* in the application program. If the result is not in **APIDWORD** format but type conversion is possible, then it is converted to **APIDWORD** format. Affects error status.

Visual Basic: If the result $\geq 2^{31}$ (2147483648), a negative value is consequently stored.

Return **APITRUE** Result exists
APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultChar, apiResultByte, apiResultInt, apiResultWord, apiResultLong, apiResultReal, apiResultText, apiResultBinary**

apiResultFormat

Summary Identifies the format of a result

**APIBOOL apiResultFormat (APIRESULTFORMAT*buf, char*result,
APIWORD set)**

buf Address of target variable in the application programme
result Result to be polled
set Result set number

Remarks Identifies the format of a result. Waits for an ongoing job to finish processing. The format of the desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* in the application program.

The following result formats are possible:

- APIFORMAT_CHAR
- APIFORMAT_BYTE
- APIFORMAT_INTEGER
- APIFORMAT_WORD
- APIFORMAT_LONG
- APIFORMAT_DWORD
- APIFORMAT_TEXT
- APIFORMAT_BINARY
- APIFORMAT_REAL

Affects error status.

Return **APITRUE** Result exists
 APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultName, apiResultNumber, apiResultSets, apiResultVar**

apiResultInt

Summary Identifies an APIINTEGER result

APIBOOL apiResultInt (APIINTEGER*buf, char*result, APIWORD set)

buf Address of target variable in the application programme
result Result to be polled
set Result set number

Remarks Identifies an **APIINTEGER** result. Waits for an ongoing job to finish processing. The desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* in the application program. If the result is not in **APIINTEGER** format but type conversion is possible, then it is converted to **APIINTEGER** format. Affects error status.

Return **APITRUE** Result exists
APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultChar, apiResultByte, apiResultWord, apiResultLong, apiResultDWord, apiResultReal, apiResultText, apiResultBinary**

apiResultLong

Summary Identifies an APILONG result

APIBOOL apiResultLong (APILONG*buf, char*result, APIWORD set)

buf Address of target variable in the application programme
result Result to be polled
set Result set number

Remarks Identifies an **APILONG** result. Waits for an ongoing job to finish processing. The desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* in the application program. If the result is not in **APILONG** format but type conversion is possible, then it is converted to **APILONG** format. Affects error status.

Return **APITRUE** Result exists
APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultChar, apiResultByte, apiResultInt, apiResultWord, apiResultDWord, apiResultReal, apiResultText, apiResultBinary**

apiResultName

Summary Identifies the result name

APIBOOL apiResultName (char*buf, APIWORD*index, APIWORD set)

buf Address of target variable in the application programme

index Result index programme

set Result set number

Remarks Identifies the result name, giving the result index and the result set. Waits for an ongoing job to finish processing. The result name is identified from the specified result index *index* (starting with 1) and the result set *set* of the result field and copied to the target variable *buf* in the application program. **APIMAXNAME** is the maximum number of copied characters and therefore the recommended buffer size of the target variables. Affects error status.

Return **APITRUE** Result exists
APIFALSE Job failed or results not present

Interface C/C++ Visual Basic DLL

See also **apiResultFormat, apiResultNumber, apiResultSets, apiResultVar**

apiResultNumber

Summary Identifies the number of results

APIBOOL apiResultNumber (APIWORDbuf*, APIWORD *set*)**

buf Address of target variable in the application programme

set Result set number

Remarks Identifies the number of results in a result set. Waits for an ongoing job to finish processing. The number of results is identified from the specified result set *set* in the result field and copied to the target variable *buf* in the application program. Affects error status.

Return **APITRUE** Result exists
 APIFALSE Job failed or results not present

Interface C/C++ Visual Basic DLL

See also **apiResultFormat, apiResultName, apiResultSets, apiResultVar**

apiResultReal

Summary Identifies an APIREAL result

APIBOOL apiResultReal (APIREAL*buf, char*result, APIWORD set)

buf Address of target variable in the application programme

result Result to be polled

set Result set number

Remarks Identifies an **APIREAL** result. Waits for an ongoing job to finish processing. The desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* in the application program. If the result is not in **APIREAL** format but type conversion is possible, then it is converted to **APIREAL** format. Affects error status.

Return **APITRUE** Result exists
APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultChar, apiResultByte, apiResultInt, apiResultWord, apiResultLong, apiResultDWord, apiResultText, apiResultBinary**

apiResultsDelete

Summary Deletes an application specific result field.

void apiResultsDelete (APIRESULTFIELD *field*)

field Reference to the result field to be deleted

Remarks -

Return -

Interface C/C++ Visual Basic DLL

See also **apiResultsNew, apiResultsScope**

apiResultSets

Summary Simplified identification of the number of result sets

APIBOOL apiResultSets (APIWORD *sets)

sets Address of target variable in the application programme

Remarks Simplified identification of the number of result sets (of the last job).
Waits for an ongoing job to finish processing. The number of result sets is identified and copied to the target variable *sets* in the application program. The result can also be identified with **apiResultWord(*sets*, "SAETZE", 0)**. Affects error status.

Return **APITRUE** Result exists
APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultFormat, apiResultName, apiResultNumber, apiResultVar**

apiResultsNew

Summary Dynamic creation of a copy of the current result field

APIRESULTFIELD apiResultsNew (void)

Remarks Dynamic creation of a copy of the current result field. Can only create one copy of the current result field. Waits for an ongoing job to finish processing. Affects error status.

Return Reference to created result field copy or **NULL** in the event of an error.

Interface C/C++ Visual Basic DLL

See also **apiResultsDelete, apiResultsScope**

apiResultsScope

Summary Switches to an application specific result field

void apiResultsScope (APIRESULTFIELD *field*)

field Reference to the result field to be switched.

Remarks Switches to an application specific result field. The lock onto the result field persists until the next job or **apiResultsScope** function call.

Return -

Interface C/C++ Visual Basic DLL

See also **apiResultsDelete, apiResultsNew**

apiResultText

Summary Identifies an APITEXT result

APIBOOL apiResultText (APITEXT*buf, char*result, APIWORD set, char*format)

buf Address of target variable in the application programme
result Result to be polled
set Result set number
format Desired format of the result (max. APIMAXNAME), a blank string must be specified to perform the default conversion.

Remarks Identifies an **APITEXT** result. Waits for an ongoing job to finish processing. The desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* in the application program. **APIMAXTEXT** is the maximum number of copied characters of the **APITEXT** type and therefore the recommended buffer size of the target variable. If the buffer size is exceeded the result is limited to **APIMAXTEXT**. The conversion instruction *format* can be used to convert the result to the **APITEXT** format. If the *format* argument is a blank string, then there is default conversion. Affects error status.

Return **APITRUE** Result exists
APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultChar, apiResultByte, apiResultInt, apiResultWord, apiResultLong, apiResultDWord, apiResultReal, apiResultBinary**

apiResultVar

Summary Identifies the name of the loaded SGBD

APIBOOL apiResultVar (APITEXT *ecu)

ecu Address of control unit variant buffer

Remarks Identifies the name of the loaded SGBD of the last job. Waits for an ongoing job to finish processing. The name is identified and copied to the target variable *ecu* in the application program. The name is stored in the system result set (result set 0) as the result VARIANTE. The result can also be identified with **apiResultText(ecu,"VARIANTE",0,"")**. The size of the **APITEXT** result copied by API can be the maximum length of the filename. Affects error status.

Return **APITRUE** Result exists
 APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultFormat, apiResultName, apiResultNumber, apiResultSets**

apiResultWord

Summary Identifies an APIWORD result

APIBOOL apiResultWord (APIWORD*buf, char*result, APIWORD set)

buf Address of target variable in the application programme

result Result to be polled

set Result set number

Remarks Identifies an **APIWORD** result. Waits for an ongoing job to finish processing. The desired result *result* is downloaded from the result set *set* in the result field and copied to the target variable *buf* in the application program. If the result is not in **APIWORD** format but type conversion is possible, then it is converted to **APIWORD** format. Affects error status.

Return **APITRUE** Result exists
APIFALSE Job failed or result not present

Interface C/C++ Visual Basic DLL

See also **apiResultChar, apiResultByte, apiResultInt, apiResultLong, apiResultDWord, apiResultReal, apiResultText, apiResultBinary**

apiSetConfig

Summary Modification of the EDIABAS configuration

APIBOOL apiSetConfig (const char **config*, const char **value*)

config Configuration element to be modified

value New configuration setting

Remarks: Modification of the configuration element *config* by specifying the new configuration setting *value*. The modification remains valid until aborted by API.

The error status is not affected.

All configuration elements and their setting are described in reference [4].

Return **APITRUE** Modified configuration

APIFALSE Erroneous modification

Interface C/C++ Visual Basic DLL

See also **apiGetConfig**

apiState

Summary Identifies the current status of job execution

int apiState (void)

Remarks Affects error status.

Return Processing status

APIBUSY Job in progress

APIREADY Job ready

APIBREAK Job aborted

APIERROR Job failed

Interface C/C++ Visual Basic DLL

See also **apiBreak, apiEnd, apilnit, apilnitExt, apiCallBack, apiSwitchDevice**

apiSwitchDevice

Summary Switches device

APIBOOL apiSwitchDevice (const chardeviceConnection*,
const char**deviceApplication*)**

deviceConnection Device connection (max. **APIMAXDEVICE**), preset connection for transmitting a blank string

deviceApplication Device application (max. **APIMAXDEVICE**), preset application for transmitting a blank string

Remarks A device change is possible by specifying the new device connection and device application. The new device is valid until **apiEnd()**, **apiInit()** or **apiInitExt()** are called. Affects error status.

Return **APITRUE** Device switched
APIFALSE Switch failed

Interface C/C++ Visual Basic DLL

See also **apiBreak, apiEnd, apiInit, apiState, apiCallBack**

5. Other Functions

Remark: The following functions are available only for MS-WINDOWS and for BMW delivery package.

enableServer

Summary Switch between server and parallel mode

BOOL enableServer (BOOL *onOff*)

onOff new mode:

TRUE Server mode
FALSE Parallel mode

Remarks Switch between server and parallel mode.

Return **TRUE** Switch is done
FALSE Switch is not done

Interface C/C++ Visual Basic DLL

See also **closeServer**

EDIABAS - API FUNCTION PRIMER

closeServer

Summary Close of EDIABAS server

void closeServer ()

Remarks Function closes the running EDIABAS server.

Return -

Interface ~~C/C++~~ Visual Basic DLL

See also **enableServer**

EDIABAS - API FUNCTION PRIMER

A. List of references

- [1] EDIABAS: Error Primer
- [2] EDIABAS: BEST/1 - Language and Interpreter
- [3] EDIABAS: Transparent Mode
- [4] EDIABAS: User Manual
- [5] EDIABAS: API Interface Description
- [6] EDIABAS: API User Manual