Error Handler

tags:
Error handling

Introduction

Access2Base uses internally an error handler that can be optionally used by users in their own code.
The principles are:

- An error has an error-level. The error-level determines the severity of the error and the behaviour of the program after the error trap: stop or go ahead ?
- All errors, whatever their level, can be registered on the request of the programmer. The technique used by Access2Base is to keep the error *traces* in memory through a *circular buffer*, i.e. a buffer where the oldest entries are replaced by the new ones when the buffer gets full.
- The error traces (circular buffer) can be viewed for debugging purposes in what is called the **Access2Base Console** dialog box.
- All the user-defined Subs or Functions follow the same code structure for error handling.

To view the errors log

- If *Access2Base* is included in the software, run next code, either directly in the Basic IDE or via a toolbar button, or ...

```
Sub Console()
        TraceConsole()
End Sub
```

- If *Access2Base* has been installed as a separate extension, then execute next menu command

> `Tools + Add-Ons + Access2Base Console ...`

in the Basic IDE.

API

The routines for error handling are:

TraceConsole	Open the *Console* dialog box for display of all past errors and logs.
TraceError	Report an error found.
TraceLevel	Set the minimal level of errors to be reported in the console.
TraceLog	Log an event in the circular buffer if its level is at least equal to the minimal error level.
DebugPrint	Record as many values as wanted in tabular form in the circular buffer, typically for debugging.

Error levels

Level	Type	Description
"DEBUG"	String	To report values of variables during the program execution. Such reported errors are NOT user visible.
"INFO"		To report any event
"WARNING"		To report some abnormal event.
"ERROR"		To report an error trapped by a user program.
"FATAL"		To report an error caused by a user program but detected by Access2Base (f.i. "Form does not exist ..." etc.).
"ABORT"		To report an error inside the Access2Base API itself. Do not use for programmer or user errors.

Recommended program structure for error handling

The example is given for a *Sub*. It is, mutatis mutandis, equally valid for a *Function*.

```
Sub mySub
        On Local Error Goto Error_Sub
        ...
Exit_Sub:
        Exit Sub
Error_Sub:
        TraceError("ERROR", Err, "MySub", Erl)
        Goto Exit_Sub
End Sub
```

In case of error next message will be displayed to the user and simultaneously registered in the trace buffer:

> Error # *number* (*...description...*) occurred at line *line* in mySub.

Bookmark this page » » Error Handler