

ibec_ImportData

ibec_ImportData function implemented. This function returns the number of imported (inserted) records.

Syntax

```
function ibec_ImportData(TargetConnection : variant; TargetTable : string;
```

```
    ImportType : integer; SourceFile : string;  
SourceTable : string;  
    Options : string; Mappings : string; CallbackBlock :  
string) : variant;
```

ibec_ImportData imports records from the specified source (SourceFile, SourceTable) into the target database (TargetConnection, TargetTable).

Please note that when using the [FixedWidths](#) option for importing data from text files with fixed column widths, the [Mappings](#) parameter is mandatory if you import data from a text file with fixed column widths. The syntax of each item in the Mappings string is:

```
<source_field_name>=<start_position>,<length>
```

IBExpert automatically produces an IBEBlock for data import directly in the [Import Data](#) window ([Block page](#)) so you can always get it there.

The ibec_ImportData function also supports import from XLSX files (Microsoft Excel). You can specify the sheet number (zero based) which you want to process, using the Sheet option for this, i.e.: Sheet=2. If the Sheet option is not specified the first sheet will be processed.

Parameters

TargetConnection	Handle of the target connection created with <code>ibec_CreateConnection</code> or <code>ibec_CreateDatabase</code> functions.
TargetTable	Name of the target table. The table must exist in the target database before importing data.
ImportType	Type of data source. Following data sources are currently supported:
<code>__impText</code>	Plain text files, comma-separated or fixed widths values.
<code>__impClipboard</code>	Windows clipboard.
<code>__impExcel</code>	MS Excel file.
<code>__impLotus123</code>	Lotus 1-2-3 file.
<code>__impQuattroPro</code>	Quattro Pro file.
<code>__impOpenOffice</code>	OpenOffice spreadsheet.
<code>__impParadox</code>	Paradox file.
<code>__impDBase</code>	dBASE file.
<code>__impAccess</code>	MS Access database.
<code>__impAdvantage</code>	Advantage table.
<code>__impDBISAM</code>	DBISAM table.
<code>__impClarion</code>	Clarion table.
<code>__impHTML</code>	HTML file.
<code>__impXML</code>	XML file.
<code>__impWAB</code>	Windows Address Book.
<code>__impVCalendar</code>	VCalendar file.

SourceFile	Path to the source file. Provide an empty string when importing data from the Windows clipboard.
SourceTable	Name of the source table if necessary (import from MS Access database).
Options	List of import options, delimited with a semicolon. The following options are currently supported:
RowFirst=<integer>	Number of the first row to be imported. Row numbering starts from 1. The default value is 1.
RowLast=<integer>	Number of the last row to be imported. By default all rows will be imported.
RowNames=<integer>	Number of rows with field names. The default value is 0.
TrimStrings	(See Example #3.)
TrimRight	This option removes trailing spaces and control characters from string values whilst importing data.
FixedWidths	Option for importing data from text files with fixed column widths. (See Example #3.)
DateOrder=DMY DYM MDY MYD YDM YMD	Order of date parts within a string representation of date values. By default the system defined order is used.
DateSeparator=<char>	Date parts separator. By default the system defined separator will be used.
TimeSeparator=<char>	Time parts separator. By default the system defined value will be used.

DateOrder= DMY DYM MDY MYD YDM YMD	Order of date parts within a string representation of date values. By default the system defined order is used.
DateSeparator= <char>	Date parts separator. By default the system defined separator will be used.
TimeSeparator= <char>	Time parts separator. By default the system defined value will be used.
DecimalSeparator= <char>	Decimal separator. By default the system defined value will be used.
ThousandSeparator= <char>	Thousand separator. By default the system defined value will be used.
CSVDelimiter= TAB SEMICOLON COMMA SPACE <char>	Delimiter for CSV-files. The default value is a semicolon (;).
RecordSeparator= CRLF CR LF	Records delimiter. The default value is CRLF.
CommitAfter= <integer>	Number of records inserted before committing. The default value is 500.
AnsiToUTF8	Converts string values to UTF8 before being inserted into the database.
Mappings	Specifies the column mappings for an import operation. Mappings string should contain a list of pairs <target_column>=<source_column>, delimited with a semicolon. It is possible to use a column name either a column index as <target_column> and <source_column>. If the Mappings options is not specified (is NULL or an empty string) each source column will be mapped to a target column by their index, i.e. the first source column will be mapped to the first target column, the second source column to the second target one, etc.
CallbackBlock	Call back IBEBlock, which will be executed for every imported record.

Example #1

```
execute ibeblock
as
begin
    MyDB = ibec_GetDefaultConnection();

    delete from test_import;
    commit;

    cbb = 'execute ibeblock (RecCount integer)
        as
        begin
            if (ibec_mod(RecCount, 100) = 0) then
                ibec_Progress('Records inserted: ' || RecCount);
            end';

    res = ibec_ImportData(MyDB, 'TEST_IMPORT', __impText,
'D:\import\test_import.csv', '',
                        'RowLast=15000; RowFirst=1001;
                        CSVDelimiter=","; DecimalSeparator=",";
DateSeparator="-"; TimeSeparator=":";
                        DateOrder=DMY; CommitAfter=1000',
                        '',
                        cbb);
    if (res is not null) then
        ibec_ShowMessage(res || ' records imported successfully.');
```

```
end;
```

Example #2

```
execute ibeblock
as
begin
    MyDB = ibec_GetDefaultConnection();

    res = ibec_ImportData(MyDB, 'TEST_IMPORT', __impAccess,
'D:\import\demo.mdb', 'CUSTOMER',
                        '',
                        'CUSTOMER_ID="Customer ID"; 2=3; 3=4',
                        '');
    if (res is not null) then
        ibec_ShowMessage(res || ' records imported successfully.');
```

```
end;
```

Example #3

```
execute ibeblock
as
begin
    ...
    sMappings = 'Field1=1,26;' +
                'Field2=27,26;' +
                'Field3=53,45';

    Res = ibec_ImportData(DB, 'MYTABLE', __impText,
                          'D:\Import\country_fixed_colnames.txt', '',
                          'RowFirst=1; RowLast=255555;
                          TrimStrings;
                          FixedWidths;', :sMappings, cbb);
    ...
end
```

From:
<https://www.ibexpert.com/docu/> - **IBExpert**

Permanent link:
https://www.ibexpert.com/docu/doku.php?id=05-ibexpert-ibeblock-functions:05-27-miscellaneous-functions:ibec_importdata

Last update: **2023/07/05 19:10**

