

OutputTo

The *OutputTo* action outputs the data located

- in a specified **form**
- in a specified **table** or **query**

tags:
Actions

to several output formats

Applies to ...

| Object | Description |
|--------------|---|
| DoCmd | Root class. When present, its name must be "DoCmd" but the object name is optional. |

or, when applicable to a table or a query, to ...

| Object | Description |
|-----------------|--|
| Database | A database object opened with the OpenDatabase or returned by the CurrentDb methods. |

SyntaxRemark

```
[DoCmd.]OutputTo(ObjectType, ObjectName, OutputFormat, OutputFile,
AutoStart, TemplateFile, Encoding, Quality)
database.OutputTo(ObjectType, ObjectName, OutputFormat, OutputFile, AutoStart,
TemplateFile, Encoding, Quality)
```

| Argument | Optional | Type or Symbol | Description | Returned value |
|---------------------|----------|--|--|--------------------|
| ObjectType | No | acOutputTable acOutputQuery acOutputForm | The type of object to output. | True if success |
| ObjectName | Yes | String | The valid name of an object of the type selected by the <i>ObjectType</i> argument. If the <i>ObjectType</i> is acOutputForm and you want to output the active form, leave this argument blank. | |
| OutputFormat | Yes | acFormatPDF acFormatODT acFormatDOC acFormatHTML acFormatODS acFormatXLS acFormatXLSX acFormatTXT | The output format, expressed as an acFormatXXX constant. If this argument is omitted, the user will be prompted for the output format. | |
| OutputFile | Yes | String | The full name, including the path, of the file you want to output the object to. If this argument is left blank, the user will be prompted for an output file name. | |
| AutoStart | Yes | Boolean | If True , specifies that you want the appropriate application to start immediately after the <i>OutputTo</i> | |

| | | | |
|--------------|-----|---|---|
| | | | action runs, with the file specified by the <i>OutputFile</i> argument opened. |
| TemplateFile | Yes | String | Meaningful only if the <i>ObjectType</i> argument is <i>acOutputTable</i> or <i>acOutputQuery</i> and the <i>OutputFormat</i> argument is HTML. Otherwise must contain the null-length string. The full name, including the path, of the template file. |
| Encoding | Yes | acUTF8Encoding | Meaningful only if the <i>ObjectType</i> argument is <i>acOutputTable</i> or <i>acOutputQuery</i> . Otherwise must be zero. |
| Quality | Yes | acExportQualityPrint acExportQualityScreen | This argument is ignored. |

The symbolic constants can be included in your code by copying and pasting next lines:

```
Global Const acOutputTable = 0
Global Const acOutputQuery = 1
Global Const acOutputForm = 2

Global Const acUTF8Encoding = 65001

Global Const acFormatPDF = "writer_pdf_Export"
Global Const acFormatODT = "writer8"
Global Const acFormatDOC = "MS Word 97"
Global Const acFormatHTML = "HTML"
Global Const acFormatODS = "calc8"
Global Const acFormatXLS = "MS Excel 97"
Global Const acFormatXLSX = "Calc MS Excel 2007 XML"
Global Const acFormatTXT = "Text - txt - csv (StarCalc)"

Global Const acExportQualityPrint = 0
Global Const acExportQualityScreen = 1
```

Remarks

- When applied to a table or a query data *DoCmd.OutputTo(...)* and *Application.CurrentDb().OutputTo(...)* are strictly equivalent.
- The designated form MUST be open before the *OutputTo* action is called.
- If the *OutputFile* argument is not equal to space, and If the output file already exists, it will be overwritten without warning. If the user is prompted for the output file (*OutputFile* = space), the user will be requested to confirm the Save.
- The *OutputFile* argument may be expressed either in the (recommended) URL format (i.e. "*file:// ...*") or by using the operating system usual syntax (e.g. "*C:\...*" in Windows).
- Outputting a table or a query to a Calc, Excel or text file goes first through the generation of a Calc document which is saved via the adequate filter.
- A text/csv file will always be encoded using the Unicode (UTF-8) character set. The choice of the field separator will depend on the locale setting for the decimal point: if it is a dot, then the field separator will be the comma, if it is a comma, then it will be the semi-colon. All text strings will be enclosed in double quotes.

Templating and styling

Templates and CSS styles can be used when the output format is HTML and when the source of the exported data is a table or a query.

The `TemplateFile` given as argument of the `OutputTo` action may contain arbitrary HTML tags, JavaScript code or any other element that is acceptable in a HTML page. Access2Base will leave them unchanged, except when a *token* is encountered. The only accepted tokens are

| Token | Description |
|--|---|
| <code><!--Template_Title--></code> | Will be replaced by the name of the table or the query. |
| <code><!--Template_Body--></code> | Will be replaced by the effective data contained in the table or the query. |

(`<!--AccessTemplate_Title-->` and `<!--AccessTemplate_Body-->` are also accepted for compatibility with MSAccess)

If the `TemplateFile` argument is absent, the default template used by Access2Base is:

```
<!DOCTYPE html>
<html>
  <head>
    <title><!--Template_Title--></title>
  </head>
  <body>
    <!--Template_Body-->
  </body>
</html>
```

To make CSS styling easy the data themselves will be generated together with next classes:

| Class | Tags | Typical usage |
|---|--|--|
| <code>dbdatatable</code> | <code><table></code> | To not confuse the data table with other tables present in the template. |
| <code>odd</code> <code>even</code> | <code><tr></code> | To display alternate colors or backgrounds.. |
| <code>firstrow</code> <code>lastrow</code> | | To define specific borders above or under the first or the last row. |
| <code>firstcol</code> <code>lastcol</code> | <code><th></code> <code><td></code> | To define specific borders on the left or the right of the first or the last column. |
| <code>char</code> <code>numeric</code> <code>date</code> <code>negative</code> <code>bool</code> <code>null</code> | <code><td></code> | To define display effects depending on the content of the table cell. Like right justifying numbers, red fonts for negative numbers, special background for empty cells, etc. |

Body output example:

```
<table class="dbdatatable">
  <caption>myQuery</caption>
  <thead>
    <tr>
```

```

<th scope="col" class="firstcol">Month</th>
<th scope="col">Dept</th>
<th scope="col">Savings</th>
<th scope="col" class="lastcol">Comments</th>
</tr>
</thead>
<tfoot>
</tfoot>
<tbody>
<tr class="odd firstrow">
<td class="char firstcol">January</td>
<td class="char">Dept1</td>
<td class="numeric">100.0</td>
<td class="char lastcol">Some comments about this department</td>
</tr>
<tr class="even">
<td class="char firstcol">February</td>
<td class="char">Dept2</td>
<td class="numeric negative">-80.5</td>
<td class="char lastcol">No comment</td>
</tr>
<tr class="odd lastrow">
<td class="char firstcol">March</td>
<td class="char">Dept3</td>
<td class="numeric">75.9</td>
<td class="char null lastcol"></td>
</tr>
</tbody>
</table>

```

String encoding

When exporting data from a table or a query, all strings will be "UTF-8 encoded". See f.i. [here](#) for more explanation.

There are however a few exceptions. Some HTML tags (= NOT ALL) found within data strings will not be encoded. Typically links to pages located in <a> tags will remain unchanged.

Next patterns will not be encoded:

| Pattern | Remark |
|------------------|---|
| " ' | |
| & | "&" will be encoded as & ; "& " will not be encoded as & ; |
| < > | |
| | Multiple normal spaces will probably appear as one single space |
| <pre> </pre> | To preserve spacing and line breaks |
| | To force a line break |
| | References to hyperlinks or anchors may be present and will be preserved in strings |

| | |
|---|---|
| <pre><a href="..." <a id="..." </pre> | |
| <pre><img src="..."</pre> | |
| <pre><span style="..." </pre> | The tag provides a way to add a specific style or other hook to a part of a text or a part of a document |
| <pre> <u> </u> <i> </i></pre> | Usual editing tags |

Error messages

| |
|---------------------------------------|
| Object '...' not found |
| Action not applicable in this context |

See also

HtmlEncode

RunApp

SelectObject

SendObject

Examples

Output form in PDF format to a file (you will be prompted for the file name and location) and open the generated file (asynchronously)

```
Const acOutputForm = 2
Const acFormatPDF = "writer_pdf_Export"
    OutputTo(acOutputForm, "myform", acFormatPDF,, True)
```

Additionally, styling examples can be found in **Output table or query to html - styling examples**.

Bookmark this page » » [OutputTo](#)