

# CSV2Code

## User Manual

Revision: 1.02a  
Date: 2016-02-15

CSV2Code can be used to generate code out of records in CSV format, as explained in this documentation.

CSV2Code is freeware. You can use it as long as you accept my Licence Agreement and Copyright (see below).

© 2013-2016 by Christian Maas – All Rights Reserved

[www.chmaas.handshake.de](http://www.chmaas.handshake.de)

[chmaas@handshake.de](mailto:chmaas@handshake.de)

If you like CSV2Code, want to appreciate my work, and/or support the development, I would be pleased to receive a donation from you via PayPal. Click to the link below to make a donation:

<http://www.chmaas.handshake.de/delphi/freeware/freeware.htm>

## **Contents**

<a href="#">1. Installation.....</a>	<a href="#">3</a>
<a href="#">2. Uninstall.....</a>	<a href="#">3</a>
<a href="#">3. Version history.....</a>	<a href="#">3</a>
<a href="#">4. Introduction.....</a>	<a href="#">4</a>
<a href="#">5. Example 1: SQL.....</a>	<a href="#">6</a>
<a href="#">6. Example 2: HTML.....</a>	<a href="#">11</a>
<a href="#">7. Example 3: Pascal code.....</a>	<a href="#">14</a>
<a href="#">8. Example 4 a: CSV (reordering and suppressing columns, change CSV separator).....</a>	<a href="#">15</a>
<a href="#">9. Example 4 b: CSV.....</a>	<a href="#">15</a>
<a href="#">10. Usage of options to generate code.....</a>	<a href="#">16</a>
<a href="#">11. Licence Agreement and Copyright.....</a>	<a href="#">17</a>

## 1. Installation

CSV2Code is a 32 bit application and requires Windows 9x/NT/2000/XP/Vista/7/8.

No special installation procedure is required, i.e. CSV2Code is a *portable application*.

Create a directory for CSV2Code, e.g. C:\CSV2Code or C:\PortableApps\CSV2Code.  
Note for Windows Vista and above: Don't create a directory below the C:\Program Files directory like C:\Program Files\CSV2Code, because in this case, the program would have no access to its own INI/TPL files.

Now simply unzip the file CSV2Code.ZIP into this directory: You'll see the following files:

CSV2Code.EXE	executable
CSV2Code.PDF	user manual
README.TXT	

No data will be written into the registry, no DLLs are copied to your hard disk. The user defined settings are stored in two files (INI and TPL) which are located in the same directory as the executable.

## 2. Uninstall

To uninstall CSV2Code completely, simply delete the directory and all containing files.

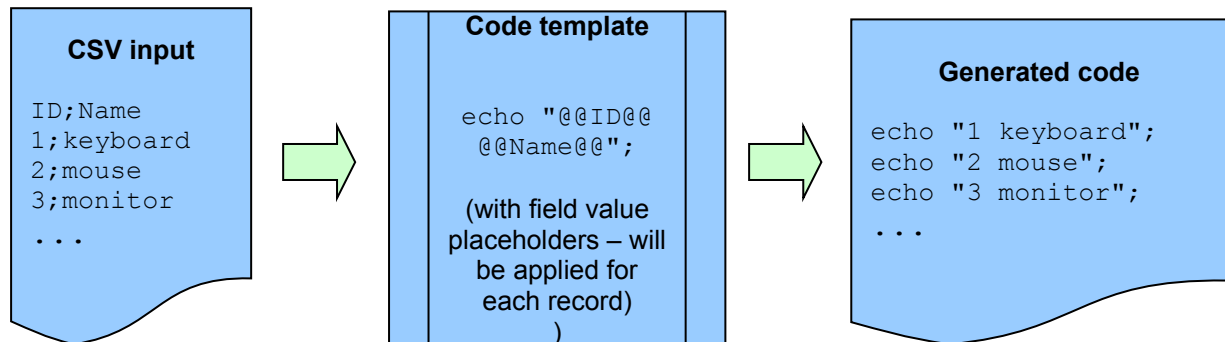
## 3. Version history

	Release date	Features/Bugfixes
1.0	2013-05-15	<ul style="list-style-type: none"><li>Initial version.</li></ul>
1.01	2014-09-03	<ul style="list-style-type: none"><li>Bugfix: When reading CSV data from file (instead of memo in GUI), I/O error 103 could occur.</li></ul>
1.02	2016-02-15	<ul style="list-style-type: none"><li>Bugfix: Empty field at the end of a line wasn't properly handled (the value of the previous field was used)</li></ul>

## 4. Introduction

### Concept

CSV2Code was developed to automate a tedious task: to generate code from records in CSV format.



The **CSV input** data must contain a **header line with the field names**.

The **code template** contains **field value placeholders**, represented by field names enclosed by a special prefix/suffix, e.g. @@ID@@ as the field value placeholder for the field named ID. The code template can consist of one or more lines.

The code template is applied to each record, thus for each record a "piece" of code (normally a single line) will be generated by **replacing all field value placeholders by the actual field values** of the record.

CSV2Code simplifies writing the code template, because it reads the CSV header beforehand and provides a context menu for inserting field value placeholders in several styles. In addition, the context menu also enables you to insert the field names itself.

**Details**

- The CSV separator is selectable (TAB, semicolon, or comma).
- The CSV data can either be pasted into a memo field, or can be read from a file.
- The program analyzes the CSV data, i.e. checks if the separator is valid, reads the field names, and builds a context menu to insert field names and field value placeholders.
- When writing the code template, you can use the context menu.
- The code will be generated with options, e.g. replacing up to three text patterns or duplicating single quotes

**The typically created code is**

- SQL statements
- HTML code
- XML code
- Programming language code (PHP, Java, C++, Pascal, ...)
- CSV (allows reordering and/or suppressing columns and/or change CSV separator)
- Whatever you need and can imagine!

Examples for three kinds of code will be presented within this documentation.

## 5. Example 1: SQL

Assume you have a table in your SQL database with records of employees. You've exported them into a spreadsheet application where you did update the department ID:

ID	LastName	FirstName	Comment	Department_ID
1	Maas	Christian	developer of CSV2Code	10
2	Doe	John	NULL	20
3	Smith	Jack	Jack's the greatest	20
4	...	...	...	...

Now you want to update the department IDs in your table, e.g. you need the following SQL statements:

**UPDATE tblEmployees SET Department\_ID = 10 WHERE ID = 1**

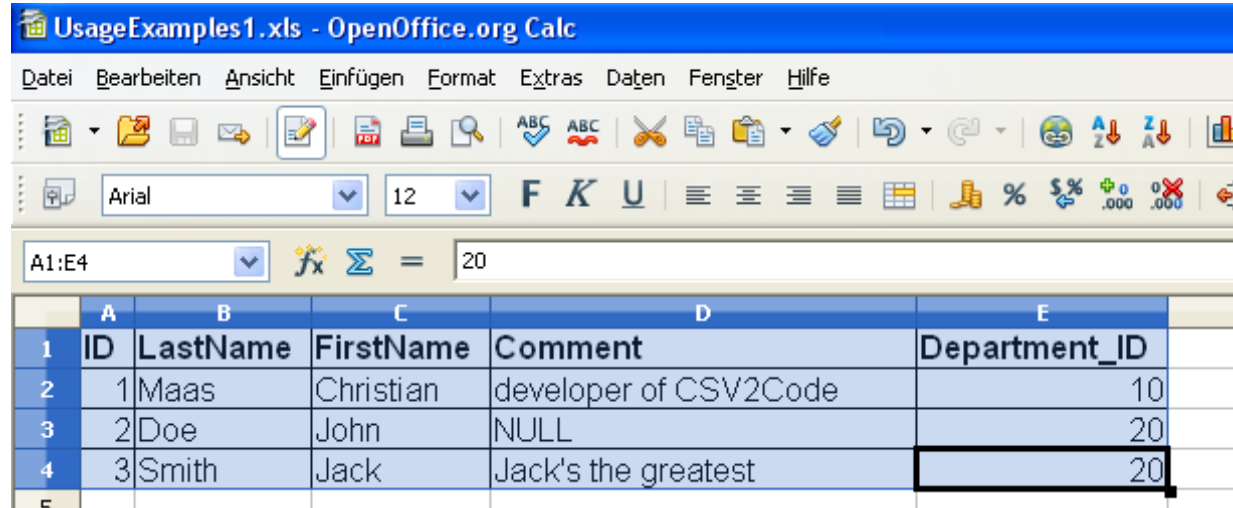
**UPDATE tblEmployees SET Department\_ID = 20 WHERE ID = 2**

**UPDATE tblEmployees SET Department\_ID = 20 WHERE ID = 3**

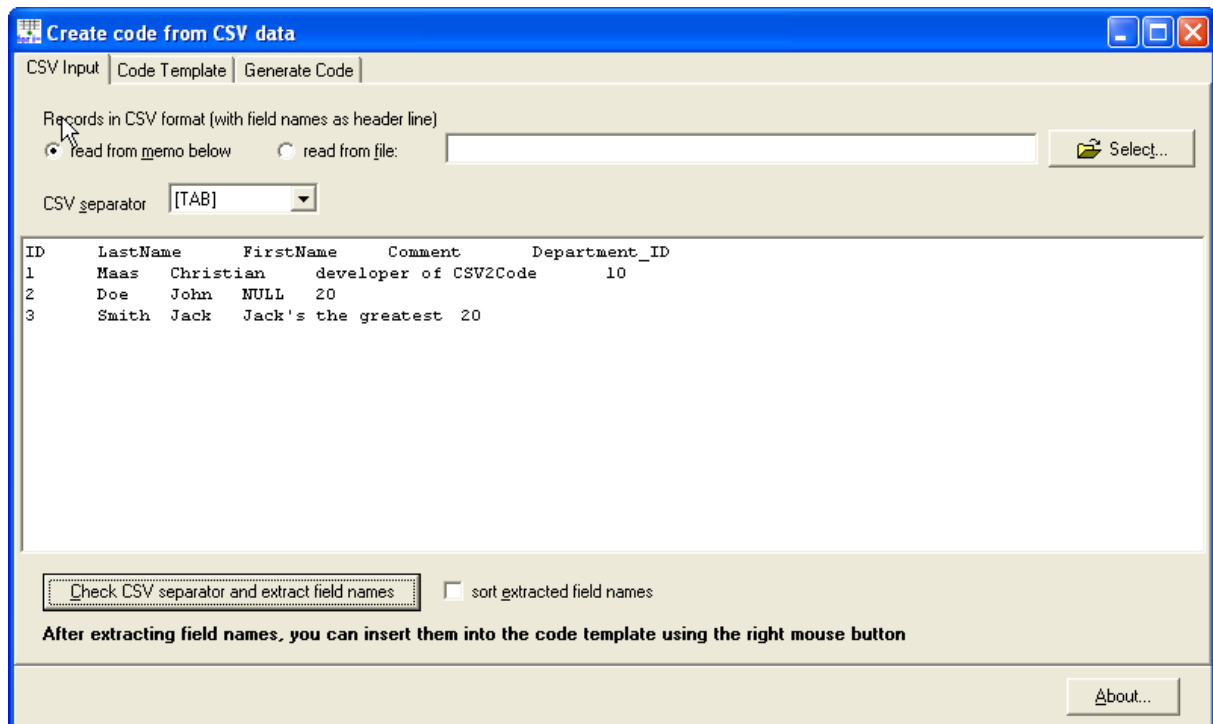
...

### Step 1: export the data from the spreadsheet application

Mark the records and copy them to the clipboard

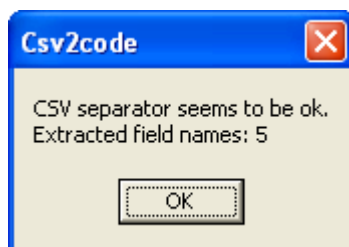


Note: You can also save the records as a CSV file.

**Step 2: paste the data into CSV2Code**

- Select the CSV separator (TAB in this example)
- check "sort extracted field names" if you want a sorted context menu to insert the field names and their placeholders
- click on "Check CSV separator and extract field names"

A message should appear that everything is ok:

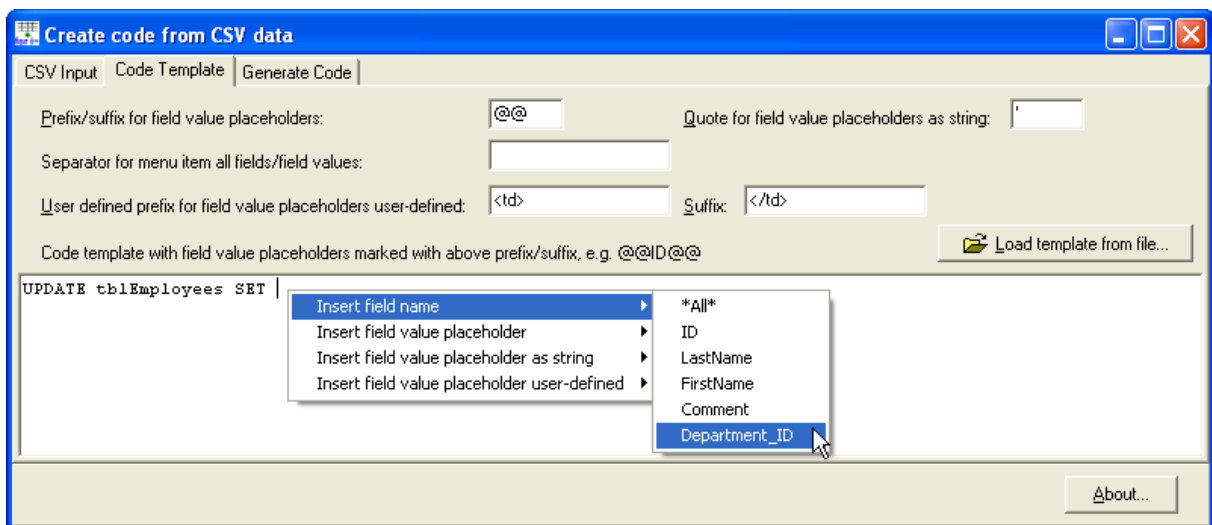


### Step 3: Writing the code template

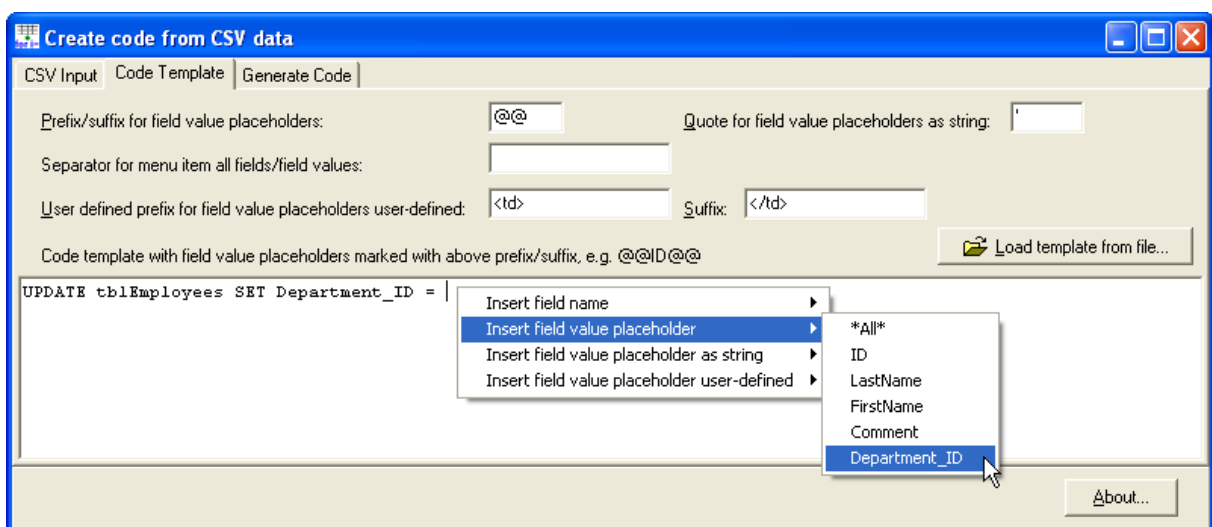
Go to the tab "Code Template". You can leave the settings "Prefix/suffix for field value placeholders" etc. Simply begin to enter the code of the template within the appropriate memo field:

```
UPDATE tblEmployees SET
```

Here you need to enter the **field name** for the department ID. The program simplifies this by providing a context menu. Click with the right mouse button at the position where the field name should be inserted and select the right field name from the context submenu "Insert field name":



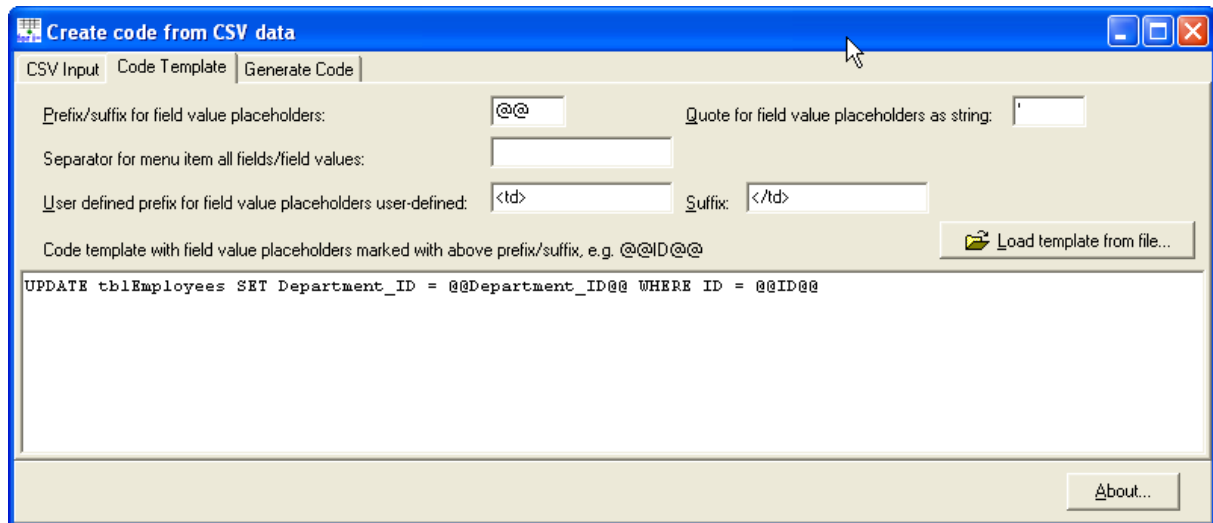
Continue entering the = sign. After the = sign, you need **the field value placeholder** for the actual contents of the field (i.e. the field value of the current record the template is applied to). The context submenu "Insert field value placeholder" makes that very easy:



Note the difference: @@Department\_ID@@ is inserted, because the *actual value of the field* is needed, not the field name.



The same way, we finally enter the WHERE-clause of the SQL statement template. The GUI should now look like this:



**The last used code template will be restored when restarting CSV2Code.**

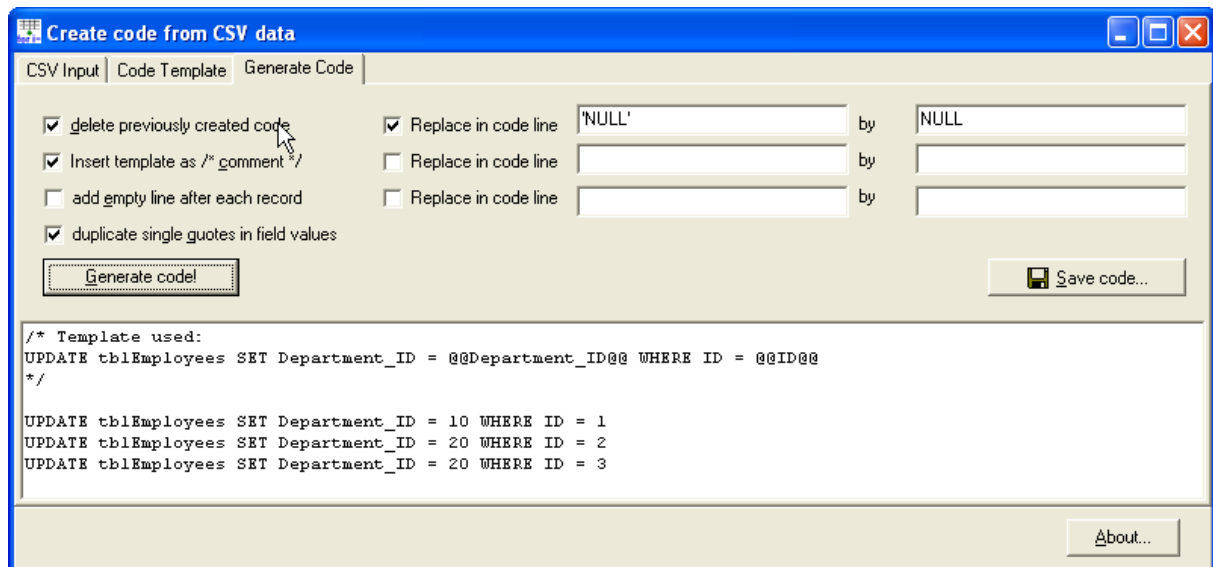
**The context menu is providing submenus to insert a**

- *field name*, e.g. ID
- *field value placeholder*, e.g. @@ID@@; the defined prefix/suffix is used to mark the placeholder (change the default @@ to whatever suits your needs)
- *field value placeholder as string*, e.g. '@@LastName@@' (the defined prefix/suffix is used, and the placeholder is enclosed within the defined quotation mark)  
Example:  
`UPDATE tblEmployees SET LastName = '@@LastName@@' WHERE ID = @@ID@@`
- *field value placeholder user-defined*, e.g. <td>@@Comment@@</td> (both the user defined prefix/suffix and the normal prefix/suffix are used)

**The special menu item \*All\* will insert all field names, field value placeholders, etc. separated by a string provided with the input field *Separator for menu item all fields/field values*.**

**Step 4: Creating the code**

Switch to the "Generate Code" tab, select the desired options and press the appropriate button:

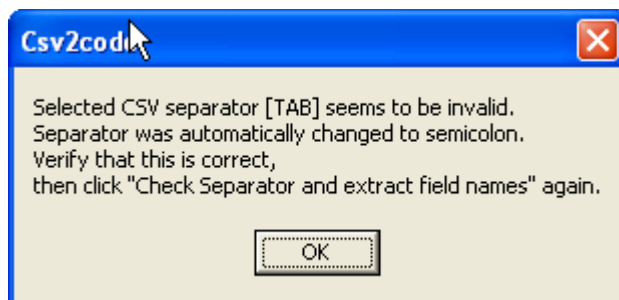
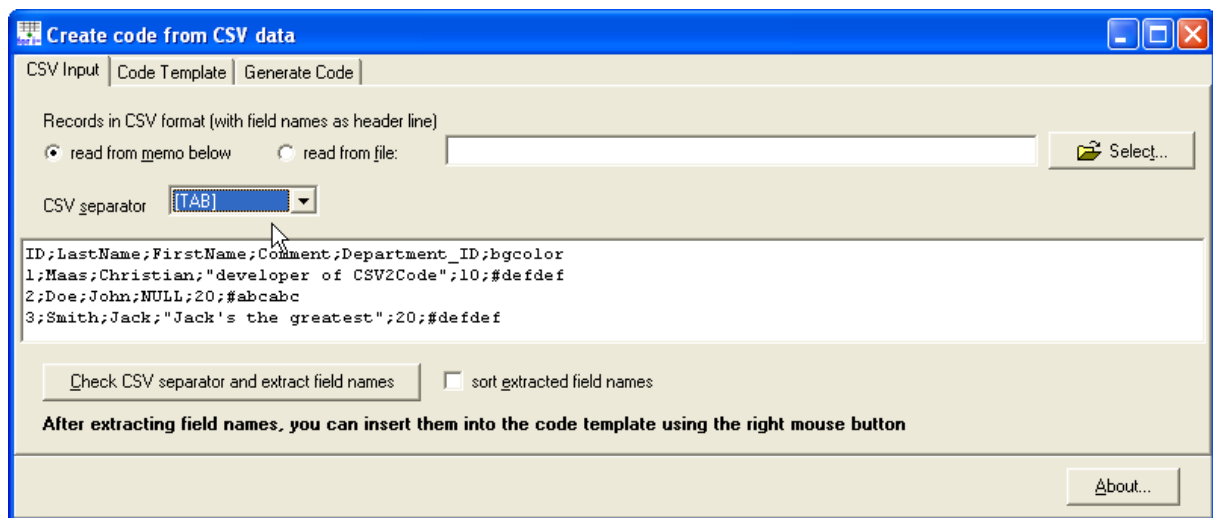


## 6. Example 2: HTML

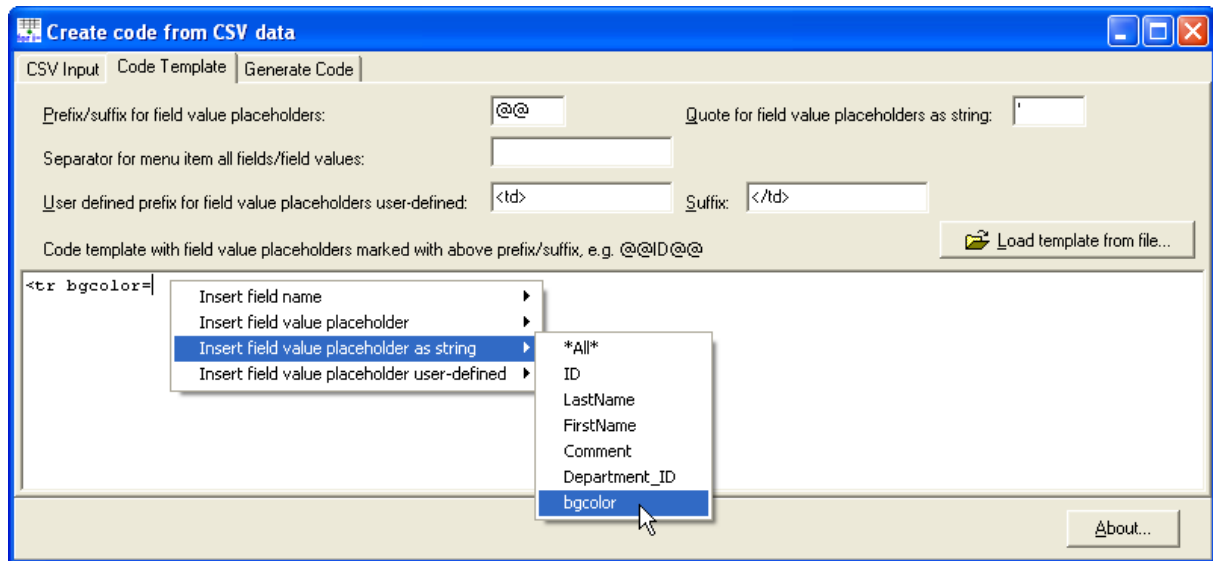
The following CSV records (separated with semicolon) are given. We want to generate code for an HTML table.

```
ID;LastName;FirstName;Comment;Department_ID;bgcolor
1;Maas;Christian;"developer of CSV2Code";10;#defdef
2;Doe;John;NULL;20;#abcabc
3;Smith;Jack;"Jack's the greatest";20;#defdef
```

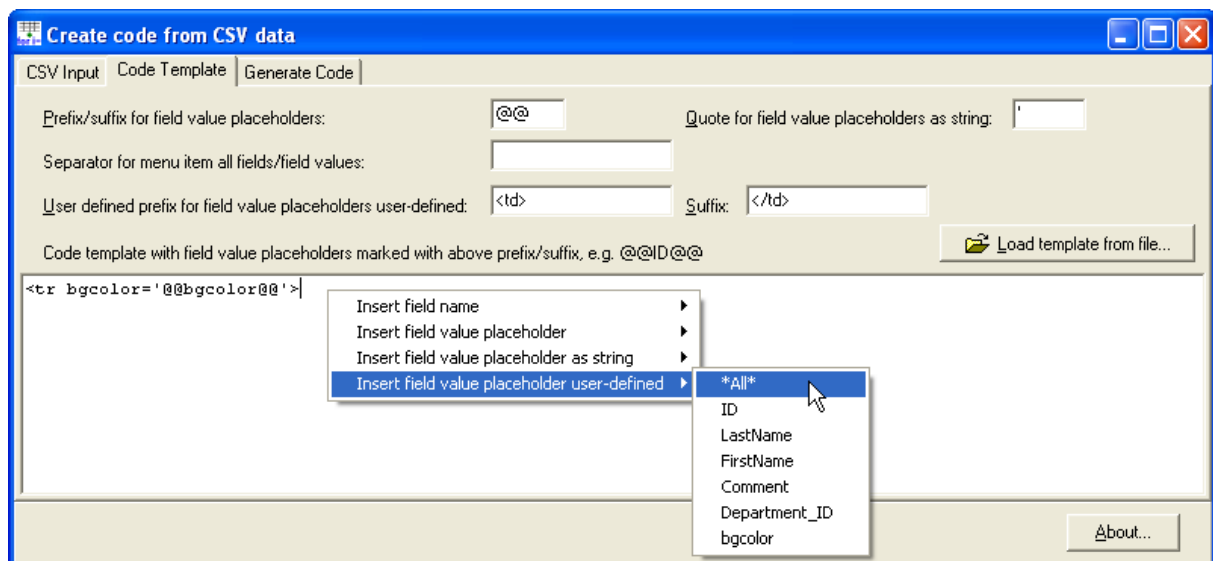
If you forget to change the previously used separator (TAB), paste this data into CSV2Code, and click the button "Check CSV separator and extract field names", the programs automatically recognizes the wrong separator:



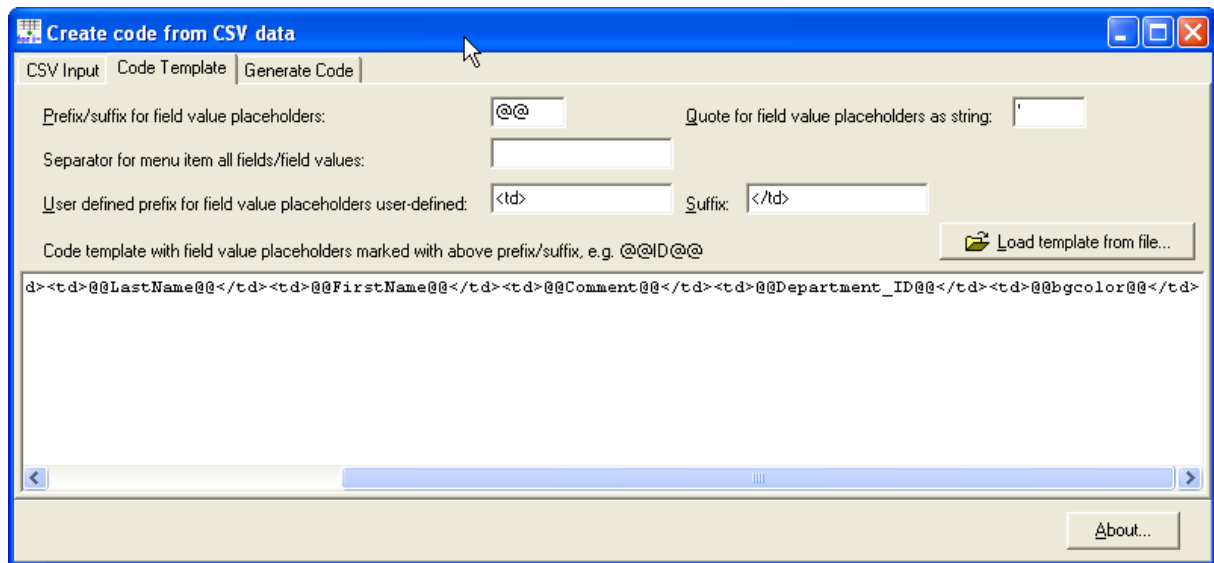
In the template, we start an HTML row with the background color taken from each record:



Terminate the table row tag with the greater sign. At this point, both the usage of the user-defined prefix/suffix as well as the menu item to insert all field value placeholders is helpful:



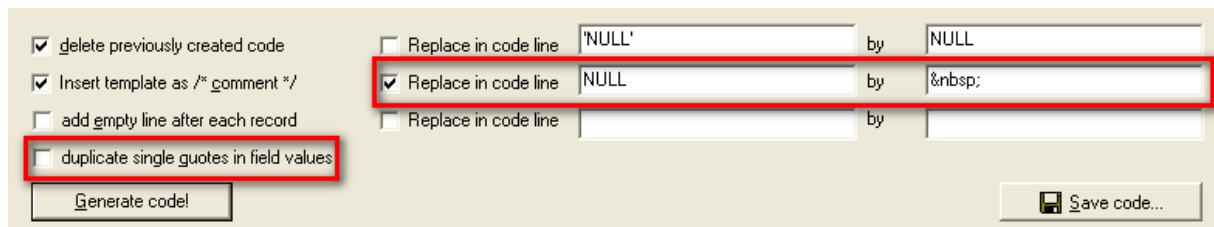
This is the result:



Delete the unnecessary last <td> entry and complete the template:

```
<tr>
bgcolor='@@bgcolor@@'><td>@@ID@@</td><td>@@LastName@@</td><td>@@FirstName@@
</td><td>@@Comment@@</td><td>@@Department_ID@@</td></tr>
```

When creating code, make sure that single quotes are not duplicated and NULL will be replaced by &nbsp;



Now you can embed the generated code

```
<tr>
bgcolor='#defdef'><td>1</td><td>Maas</td><td>Christian</td><td>developer of
CSV2Code</td><td>10</td></tr>
<tr>
bgcolor='#abcabc'><td>2</td><td>Doe</td><td>John</td><td>&nbsp;</td><td>20<
/td></tr>
<tr bgcolor='#defdef'><td>3</td><td>Smith</td><td>Jack</td><td>Jack's the
greatest</td><td>20</td></tr>
```

into HTML. This is the result shown in a browser:

1	Maas	Christian	developer of CSV2Code	10
2	Doe	John		20
3	Smith	Jack	Jack's the greatest	20

## 7. Example 3: Pascal code

We have CSV records containing an error code and an error description:

```
ErrorCode;ErrorDescription
100;"Disk read error"
101;"Disk write error"
102;"File not assigned"
103;"File not open"
104;"File not open for input"
105;"File not open for output"
```

This is our code template (which does **consist of several lines**) to generate Pascal code:

```
if ErrorCode = @@ErrorCode@@ then begin
    ShowMessage ('@@ErrorDescription@@');
    Exit;
end;
```

This creates the following code (the option "add empty line after each record" is checked):

```
if ErrorCode = 100 then begin
    ShowMessage ('Disk read error');
    Exit;
end;

if ErrorCode = 101 then begin
    ShowMessage ('Disk write error');
    Exit;
end;

if ErrorCode = 102 then begin
    ShowMessage ('File not assigned');
    Exit;
end;

if ErrorCode = 103 then begin
    ShowMessage ('File not open');
    Exit;
end;

if ErrorCode = 104 then begin
    ShowMessage ('File not open for input');
    Exit;
end;

if ErrorCode = 105 then begin
    ShowMessage ('File not open for output');
    Exit;
end;
```

## 8. Example 4 a: CSV (reordering and suppressing columns, change CSV separator)

The records are the same like in our HTML example:

```
ID;LastName;FirstName;Comment;Department_ID;bgcolor
1;Maas;Christian;"developer of CSV2Code";10;#defdef
2;Doe;John;NULL;20;#abcabc
3;Smith;Jack;"Jack's the greatest";20;#defdef
```

We want to create a CSV file with only three columns (LastName, Comment, ID) which are separated by TAB. For this purpose, our code template is

```
@@LastName@@      @@Comment@@ @@ID@@
```

Note that there are TABs between the field value placeholders; you can simply enter them within the code template memo field.

With "duplicate single quotes in field values" *unchecked*, the resulting code is

```
Maas  developer of CSV2Code  1
Doe   NULL  2
Smith Jack's the greatest    3
```

When converting semicolon or comma separated records into TAB separated ones, double quotes are removed automatically.

## 9. Example 4 b: CSV

When converting TAB separated records into semicolon or comma separated ones, you must ensure that double quotes are generated if necessary. Example:

CSV records (TAB separated): ..

```
LastName → Comment → ID¶
Maas → developer of CSV2Code → 1¶
Doe → NULL → 2¶
Smith → Jack's the greatest → 3¶
```

Code template:

```
@@ID@@;"@@LastName@@";"@@Comment@@"
```

Generated Code:

```
1;"Maas";"developer of CSV2Code"
2;"Doe";"NULL"
3;"Smith";"Jack's the greatest"
```

## 10. Usage of options to generate code

Most options are self-explaining. The following are explained in detail.

### **Insert template as */\* comment \*/***

This can make it easier to repeat the same task.

### **Duplicate single quotes in field values**

Use this option if the code uses the single quote as string terminator. In our SQL example, the third record contains a single quote in the comment field. For the code template

```
UPDATE tblEmployees SET Comment = '%%Comment%%' WHERE ID = %%ID%%
```

will be generated *with this option checked*:

```
UPDATE tblEmployees SET Comment = 'Jack''s the greatest' WHERE ID = 3
```

But without this option, invalid SQL code will be generated:

```
UPDATE tblEmployees SET Comment = 'Jack's the greatest' WHERE ID = 3
```

### **String replacement**

You can replace up to three text patterns. This is useful e.g. in our SQL example above, where the comment field of the second record contains NULL. Because the comment field has the string type, its field value placeholder must be enclosed in quotes within the code template, e.g.

```
UPDATE tblEmployees SET Comment = '%%Comment%%' WHERE ID = %%ID%%
```

Without replacing 'NULL' by NULL, the following code will be generated:

```
UPDATE tblEmployees SET Comment = 'NULL' WHERE ID = 2
```

Normally, the field should have the NULL value instead of the string "NULL":

```
UPDATE tblEmployees SET Comment = NULL WHERE ID = 2
```



## 11. Licence Agreement and Copyright

### IMPORTANT - READ CAREFULLY

This license and disclaimer statement constitutes a legal agreement ("License Agreement") between you (either as an individual or a single entity) and Christian Maas (the "Author") for this software product ("Software"), including any software, media, and accompanying on-line or printed documentation.

BY DOWNLOADING, INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE, YOU AGREE TO BE BOUND BY ALL OF THE TERMS AND CONDITIONS OF THIS LICENSE AND DISCLAIMER AGREEMENT.

1. This software is freeware. You can use this software royalty-free for private and commercial purposes.
2. You can freely distribute copies of the main archive as long as no alterations are made to the contents and no charge is raised except a reasonable fee for distributing costs.
3. You may not modify, reverse engineer, decompile, or disassemble the object code portions of this software.
4. This Software is owned by Christian Maas and is protected by copyright law and international copyright treaty. Therefore, you must treat this Software like any other copyrighted material (e.g. a book).
5. This software is provided "as is" and without any warranties expressed or implied, including, but not limited to, implied warranties of fitness for a particular purpose.
6. In no event shall the author be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use this software or documentation, even if the author has been advised of the possibility of such damages.
7. Any feedback given to the author will be treated as non-confidential. The author may use any feedback free of charge without limitation.

© 2013-2016 by Christian Maas - All Rights Reserved