

Article

[Eduard Lebedyuk](#) · Oct 3, 2016 3m read

Tips & Tricks - SQL to Excel

End users sure love Excel tables. So, how can you deliver the second to the first? Read this post to find out.

Let's say you wrote some SQL query, and now you need to make the results look presentable. So, Excel. Excel can interpret HTML tables and display them as usual if they have an XLS extension. Here's the [code](#), and it does the following:

- Executes SQL
- Iterates over the resultset writing it to stream in a HTML format
- Writes stream to file

Snippets

You can pass any number of arguments into the entry point and they in turn would be passed into statement. The following method works as expected and is a valid Caché ObjectScript code:

```
ClassMethod generateStreamFromSQL(sql As %String, args...) As %Status
{
    set st = ##class(%SQL.Statement).%New()
    set sc = st.%Prepare(sql)
    set result = st.%Execute(args...)
}
```

Example

Call like this:

```
w ##class(Utils.XLS).generateFileFromSQL("D:\1.xls", "SELECT 123456789100+1 AS Num, T
O_CHAR(123456789101) AS Text, NOW() As DateTime, TO_CHAR(NOW()) As DateTimeText")
```

Excel displays the file like this:

	A	B	C	D
1	Num	Text	DateTime	DateTimeText
2	1,23457E+11	123456789101	03.10.2016	2016-10-03 17:25:54

And the contents of 1.xls looks like this:

```
<html>
  <head>
```

```
<style>
  .num {
    mso-number-format:General;
  }
  .text{
    mso-number-format:"\@";/*force text*/
  }
  .date {
    mso-number-format:"Short Date";
  }
</style>
</head>
<body>
  <table border="1">
    <tr style="font-weight:bold">
      <td>Num</td>
      <td>Text</td>
      <td>DateTime</td>
      <td>DateTimeText</td>
    </tr>
    <tr>
      <td class="num">123456789101</td>
      <td class="text">123456789101</td>
      <td class="date">2016-10-03 17:25:54</td>
      <td class="text">2016-10-03 17:25:54</td>
    </tr>
  </table>
</body>
</html>
```

Note the css styles - they allow proper number/string/date formatting, here are [some other values](#) for mso-number-format.

And here's an example with several arguments:

```
w ##class(Utils.XLS).generateFileFromSQL("D:\1.xls", "SELECT ID FROM Sample.Person WHERE ID = ? OR ID = ?", 1,3)
```

Links

- [GitHub](#)

[#Caché](#) [#Code Snippet](#) [#SQL](#) [#Tips & Tricks](#)

Source URL:<https://community.intersystems.com/post/tips-tricks-sql-excel>