SQL Server Management Studio (SSMS) 2012 Tips & Tricks

Craig Buchanan

Version 0.0.2
2015-10-06 08:51:55 CDT

Table of Contents

Summary

A reference to make using SSMS 2012 more productive.

Contributing

- Project is hosted at https://github.com/craibuc/ssms2012

Shortcuts

- Run query: `F5`
- Hide/show Results/Messages pane: `Ctrl+R`
- Display Messages pane: `F6`
- Go to available-servers list: `Ctrl+U`
- Show explain plan: `Ctrl+L`
- SQL Server Management Studio Keyboard Shortcuts

Snippets

Snippets are a fast way to insert large blocks of code with simple keystrokes.

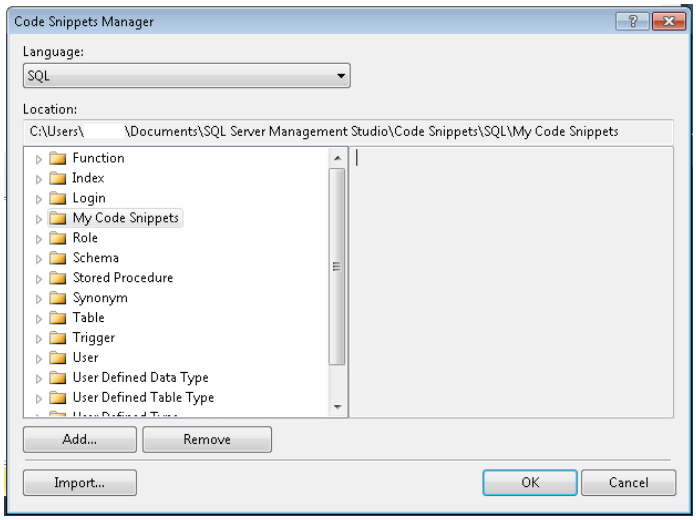
General requirements:

- Application-defined snippets are located in `C:\Program Files (x86)\Microsoft SQL Server\110\Tools\Binn\ManagementStudio\SQL\Snippets\1033` or one of its children. Non-standard subfolders (i.e. those made by the user) are ignored.
- User-defined snippets may be added to `C:\Users\<user>\Documents\SQL Server Management Studio\Code Snippets\SQL\My Code Snippets`. A remote folder may be used instead by creating a symlink: `C:\Users\<user>\Documents\SQL Server Management Studio\Code Snippets\SQL> mklink /d "My Code Snippets" "\\server\path\to\folder"`.
- File must use a valid XML document that complies with its schema (when in doubt, use an existing snippet as a template).
- File must end with `.snippet`.

Code Snippets Manager

Snippets can be managed by using the Code Snippets Manager.

Open the Code Snippets Manager by choosing `Tools | Code Snippets Manager` or by pressing `Ctrl`+`K`, `Ctrl`+`X`:



Expansion Snippets

Expansion snippets are block of code added to the cursor's location.

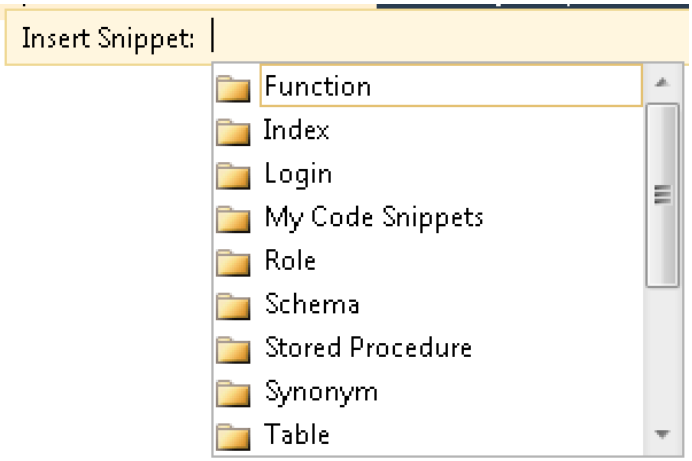
Reguirements

- Snippet definition is type `<SnippetType>Expansion</SnippetType>`

Usage

When editing a query:

- position the cursor in the desired location
- invoke by pressing `Ctrl`+`K`, `Ctrl`+`X`



- use the arrow keys to selected the desired folder and snippet
- press `Enter`

Examples

TODO

Surround Snippets

Requirements

- Snippet definition is type `<SnippetType>SurroundsWith</SnippetType>`.

Usage

When editing a query:

- highlight the desired text
- invoke by pressing `Ctrl`+`K`, `Ctrl`+`S`
- use the arrow keys to selected the desired folder and snippet
- press `Enter`

Examples

Milliseconds

Removes milliseconds from a date/time field.

```
<?xml version="1.0" encoding="utf-8" ?>
<CodeSnippets  xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
<_locDefinition xmlns="urn:locstudio">
<_locDefault _loc="locNone" />
<_locTag _loc="locData">Title</_locTag>
<_locTag _loc="locData">Description</_locTag>
<_locTag _loc="locData">Author</_locTag>
<_locTag _loc="locData">ToolTip</_locTag>
<_locTag _loc="locData">Default</_locTag>
</_locDefinition>
<CodeSnippet Format="1.0.0">
<Header>
<Title>Milliseconds</Title>
<Shortcut></Shortcut>
<Description>Removes milliseconds from a date/time field.</Description>
<SnippetTypes>
<SnippetType>SurroundsWith</SnippetType>
</SnippetTypes>
</Header>
<Snippet>
<Declarations>
<Literal>
<ID>Alias</ID>
<ToolTip>The alias for the expression.</ToolTip>
<Default>SOME_TIME</Default>
</Literal>
</Declarations>
<Code Language="SQL">
<![CDATA[CONVERT(VARCHAR, $selected$, 120) [$Alias$]]]>
</Code>
</Snippet>
</CodeSnippet>
</CodeSnippets>
```

Pivot

Inserts basic Pivot structure.

```
<?xml version="1.0" encoding="utf-8" ?>
<CodeSnippets  xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
<_locDefinition xmlns="urn:locstudio">
<_locDefault _loc="locNone" />
<_locTag _loc="locData">Title</_locTag>
<_locTag _loc="locData">Description</_locTag>
<_locTag _loc="locData">Author</_locTag>
<_locTag _loc="locData">ToolTip</_locTag>
<_locTag _loc="locData">Default</_locTag>
</_locDefinition>
<CodeSnippet Format="1.0.0">
<Header>
<Title>Pivot</Title>
<Shortcut></Shortcut>
<Description>Code Snippet for a Pivot.</Description>
<SnippetTypes>
<SnippetType>SurroundsWith</SnippetType>
</SnippetTypes>
</Header>
<Snippet>
<Declarations>
<Literal>
<ID>Measure</ID>
<ToolTip>Field to be summarized</ToolTip>
<Default>Measure</Default>
</Literal>
<Literal>
<ID>Pivot</ID>
<ToolTip>Field to be pivoted</ToolTip>
<Default>Pivot</Default>
</Literal>
<Literal>
<ID>Value</ID>
<ToolTip>Column value</ToolTip>
<Default>Value</Default>
</Literal>
</Declarations>
<Code Language="SQL">
<![CDATA[
PIVOT (
MAX($Measure$)
FOR $Pivot$ IN ([$Value$],[$Value$])
) p
]]>
</Code>
</Snippet>
</CodeSnippet>
```

```
</CodeSnippets>
```

Stuff

Inserts block to serialize (semi-colon delimited) a field.

```
<?xml version="1.0" encoding="utf-8" ?>
<CodeSnippets  xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
<_locDefinition xmlns="urn:locstudio">
<_locDefault _loc="locNone" />
<_locTag _loc="locData">Title</_locTag>
<_locTag _loc="locData">Description</_locTag>
<_locTag _loc="locData">Author</_locTag>
<_locTag _loc="locData">ToolTip</_locTag>
<_locTag _loc="locData">Default</_locTag>
</_locDefinition>
<CodeSnippet Format="1.0.0">
<Header>
<Title>Stuff</Title>
<Shortcut></Shortcut>
<Description>Inserts block to serialize (semicolon delimited) a field.</Description>
<SnippetTypes>
<SnippetType>SurroundsWith</SnippetType>
</SnippetTypes>
</Header>
<Snippet>
<Declarations>
<Literal>
<ID>Field</ID>
<ToolTip>The field to be serialized.</ToolTip>
</Literal>
<Literal>
<ID>Alias</ID>
<ToolTip>The alias for the expression.</ToolTip>
</Literal>
</Declarations>
<Code Language="SQL">
<![CDATA[
STUFF((
SELECT '; ' + $Field$
FROMTABLE t
WHEREKEY = KEY
ORDER BY $Field$
FOR XML PATH(''),TYPE).value('(./text())[1]','VARCHAR(MAX)'
),1,2,'') AS $Alias$
]]>
</Code>
</Snippet>
</CodeSnippet>
</CodeSnippets>
```

Varchar

Converts a field to a VARCHAR.

```
<?xml version="1.0" encoding="utf-8" ?>
<CodeSnippets  xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
<_locDefinition xmlns="urn:locstudio">
<_locDefault _loc="locNone" />
<_locTag _loc="locData">Title</_locTag>
<_locTag _loc="locData">Description</_locTag>
<_locTag _loc="locData">Author</_locTag>
<_locTag _loc="locData">ToolTip</_locTag>
<_locTag _loc="locData">Default</_locTag>
</_locDefinition>
<CodeSnippet Format="1.0.0">
<Header>
<Title>Varchar</Title>
<Shortcut></Shortcut>
<Description>Converts a field to a VARCHAR.</Description>
<SnippetTypes>
<SnippetType>SurroundsWith</SnippetType>
</SnippetTypes>
</Header>
<Snippet>
<Declarations/>
<Code Language="SQL">
<![CDATA[CAST($selected$ AS VARCHAR)]]>
</Code>
</Snippet>
</CodeSnippet>
</CodeSnippets>
```

SQL Command Mode

`SQL CMD Mode` adds simple macros to the development environment. It is also available from the command line; see SQLCMD .

Start SQL Command Mode by choosing `Query | SQL CMD Mode`.

Reuse Parameters

The scope of the `DECLARE` statement is one code block:

```
-- ensure that DECLARE in earlier section of code matches these values
DECLARE @STARTING_DATE DATE = '09/01/15';
DECLARE @ENDING_DATE DATE = '09/30/15';

-- perform the first query, using the specified date range, producing the first result
set
SELECT  *
FROM    TABLE_0
WHERE   STARTING_DATE >= @STARTING_DATE AND ENDING_DATE <= @ENDING_DATE
GO

-- ensure that DECLARE in earlier section of code matches these values
DECLARE @STARTING_DATE DATE = '09/01/15';
DECLARE @ENDING_DATE DATE = '09/30/15';

-- perform a second query, using the same date range, producing the second result set
SELECT  *
FROM    TABLE_1
WHERE   STARTING_DATE >= @STARTING_DATE AND ENDING_DATE <= @ENDING_DATE
GO
```

In SQL Command Mode, variables can be reused:

```
:setvar starting_date '09/01/15'
:setvar ending_date '09/30/15'

-- perform the first query, using the specified date range, producing the first result
set
SELECT  *
FROM    TABLE_0
WHERE   STARTING_DATE >= $(starting_date) AND ENDING_DATE <= $(ending_date)
GO

-- perform a second query, using the same date range, producing the second result set
SELECT  *
FROM    TABLE_1
WHERE   STARTING_DATE >= $(starting_date) AND ENDING_DATE <= $(ending_date)
GO
```

Automatic exports

Automatically export a result set as a CSV file.

```
-- disable counts; '(x row(s) affected)' message that is echoed to file
SET NOCOUNT ON
GO

-- define a base path that can be repeatedly referenced, if necessary
:setvar path c:\path\to\directory\

-- perform a query, saving the results to a CSV file
:Out $(path)TABLE_0.csv
SELECT  *
FROM    TABLE_0
GO

-- restore counts
SET NOCOUNT OFF
GO
```

Gotchas

Exporting CSV Files

Null Values

When exporting a query's results to a CSV file, SSMS 2012 substitutes the word 'NULL' for `NULL` values.

```
SELECT  'LOREM IPSUM' TEXT_FIELD_0
        ,NULL NULL_FIELD
        ,'LOREM IPSUM' TEXT_FIELD_1
```

Desired:

```
TEXT_FIELD_0,NULL_FIELD,TEXT_FIELD_1
LOREM IPSUM,,LOREM IPSUM
```

Actual:

```
TEXT_FIELD_0,NULL_FIELD,TEXT_FIELD_1
LOREM IPSUM,NULL,LOREM IPSUM
```

Solution

Post-process the file to remove the word 'NULL'.

Date/Time Values

When `DATETIME` values are exported as CSV, milliseconds are included. By default, Excel will format this data as a time value, potentially leading to confusion.

```
SELECT  GetDate() MILLISECONDS
```

```
MILLISECONDS
2015-09-25 10:57:53.740
```

Solution

`CONVERT` the field to `ODBC Canonical` format. Use the Milliseconds snippet.

```
SELECT  CONVERT(VARCHAR, GetDate(), 120) NO_MILLISECONDS
```

```
NO_MILLISECONDS
2015-09-25 10:57:53
```

:Out Command

When using the `:out` macro in SQL Command Mode , textual values aren't automatically escaped with double quotes. If the text contains a comma, it will be seen as a delimiter, leading to errorenous parsing.

```
SELECT  'STRING, WITH A COMMA' FIELD_WITH_A_COMMA
```

```
FIELD_WITH_A_COMMA
STRING, WITH A COMMA
```

Solution

Use the `QuoteName` function to wrap the field in double quotes,

```
SELECT  QuoteName('STRING, WITH A COMMA', Char(34)) FIELD_WITH_A_COMMA
```

```
FIELD_WITH_A_COMMA
"STRING, WITH A COMMA"
```

Appendix

Closely-related topics and technologies.

SQLCMD

Use `sqlcmd` to run queries at the command line. See sqlcmd Utility.

Parameters

`-E`
: Use trusted connection (default)

`-S`
: Server name

`-d`
: Database name

`-v`
: Set a variable's value

```
sqlcmd -v MyVar1=something -v MyVar2="some thing"
```

`-i`
: File containing the query to execute

`-o`
: File to contain the results of the query

`-s`
: Column separator (default is ` `)

`-W`
: Remove trailing spaces from a column

Example

```
sqlcmd -E -S server -d database -i c:\users\<user>\desktop\query.sql -o
c:\users\<user>\desktop\output.csv -s ',' -W
```

T-SQL

Communicating State

Use `RAISERROR` to add information about a query's state during execution to the Messages pane.

When the query is executing, press `F6` to switch to Messages pane.

```
RAISERROR( 'Processing query 0...',0,1) WITH NOWAIT
GO

-- query 0

RAISERROR( 'Processing query 1...',0,1) WITH NOWAIT
GO

-- query 1
```

WARNING | This tends to conflict with SQL Command Mode.

Revisions

Version	Change
0.0.2	Adding surrounds content; Adding `Revisions` section; Refactoring documents
0.0.1	Document created