Ben Pink
Clayton Rath
David Vegter
Demetrios Green

Coding Standards

C#/.NET 5.0 Standards:

1. Max of 100 characters per line

   Ex:

   ```
   //This is a test message where the length has exceeded 100 character in lenght and should have been continued onto the next line.

   //This is s test message that has been managed to expand onto the next line
   //so that it can easily be read on many systems of varying screensize.
   ```

2. No trailing whitespace

   Ex:

   ```
   //This is what trailing whitespace looklike
   //And you should not do it
   ```

3. Use pascal case when naming a class, method, public fields, and constants.

Ex:

```
public class TestClass {

    public int TestNumber;

    public bool TestMethod(int input){
        TestNumber = input;

        return true;
    }
}
```

4. Use camel casing when naming private or internal fields, and prefix them with '_'.

```
class ClassExample
{

    0 references
    public void ExampleMethod()
    {
        int testNumber;
    }
}
```

5. Place the comments on a separate line and begin them with an uppercase letter.

```
//Example comments
0 references
public void ExampleMethod()
{
    // Example comment about testNumber
    int testNumber;
}
```

6. When writing method parameters, use camel casing.

Ex.

```
public class TestClass
{
    public void TestMethod(int numberOne, int numberTwo)
    {
        return numberOne * numberTwo;
    }
}
```

7. Use Allman style braces, where each brace begins on a new line.

Ex.

```
public class TestClass
{
    public void TestMethod(int numberOne, int numberTwo)
    {
        return numberOne * numberTwo;
    }
}
```

8. Use four spaces, not tabs, when indenting.

```
//Example comments
0 references
public void ExampleMethod()
{
    // Example comment about testNumber
    int testNumber;
}
```

9. Avoid more than one empty line at any time.

Ex.

```
public class TestClass
{
    public void TestMethod(int numberOne, int numberTwo)
    {
        return numberOne * numberTwo;
    }
}
```

10. Use parentheses to make clauses in an expression apparent.

```
public void ExampleMethod()
{
    int i = 0;
    int j = 0;
    if ( (i + 1) != (j + 1) )
        j = i;
}
```

SQL Standards

1. Use snake case (do not use CamelCase)

Ex: file_name

Not: File_Name or FileName

2. Only letters, numbers, and underscores should be used in names.

Ex: home_address

Not: home.address

3. SQL keywords and reserved words: uppercase –

Ex: "SELECT"

4. Identifiers (table names, column names, etc.): lowercase

Ex: "SELECT person"

5. New line should be used after each keyword call

Ex: "SELECT *

   FROM person"

6. Use spaces, not tabs.

Ex: "SELECT person

   WHERE age > 9

     AND height > 6"

Not: "SELECT person

       WHERE age > 9

        AND height > 6"

7. Lines must have no trailing spaces.

Ex: Do this: "SELECT *"

   Not this: "SELECT * _____ "

8. Primary keys should be unique, and the key data type should be consistent across tables.

Ex: "create table tutorials_tbl(

        PRIMARY KEY (tutorial_id));

9. If an SQL statement is long, add line breaks to reformat it and make it easier to read.

```
1   SELECT f1 AS "my_column", f10 ....
2   FROM mysqltest1.t5
3   WHERE (f2 BETWEEN 17 AND 25 OR f2 = 61)
4     AND f3 IN (SELECT ....
5                  FROM mysqltest1.t4
6                  WHERE .....)
7   ORDER BY ...
```

10. Header of Scripts

-For each test or auxiliary script, create a header that contains the following information:

- Purpose of the test or script
- Corresponding WL task, if there is any
- Creator of the test and date of creation
- Author of last significant change + date of change + what was changed
- Dates should be in ISO format (ISO 8601): YYYY-MM-DD
- In case that the script assigns values to some variables and sources some master test script, please explain the purpose of these variables and why you use these values

Ex:

```
########## suite/funcs_1/t/processlist_val_no_prot.test #############
#                                                                    #
# Testing of values within INFORMATION_SCHEMA.PROCESSLIST            #
#                                                                    #
# The prepared statement variant of this test is                    #
# suite/funcs_1/t/processlist_val_ps.test.                          #
#                                                                    #
# There is important documentation within                           #
#         suite/funcs_1/datadict/processlist_val.inc                #
#                                                                    #
#                                                                    #
# Creation:                                                          #
# 2007-08-09 mleich Implement this test as part of                  #
#                   WL#3982 Test information_schema.processlist      #
#                                                                    #
######################################################################
```