Copyright 2010 Enterim LLC Paul M Drumm

Notes on Bugzilla Installation to make work on MS SQL 2005 or 2008

If installing on MSSQL2005 the also install the 2008 native client.

Download latest active state (5.10.1 build 1006) version and install
http://downloads.activestate.com/ActivePerl/Windows/5.10/ActivePerl-5.10.1.1006-MSWin32-x86-291086.msi

Download latest Bugzilla (3.4.2) and install
http://ftp.mozilla.org/pub/mozilla.org/webtools/bugzilla-3.4.2.tar.gz

Run the following from the bugzilla directory.

```
ppm install TimeDate
ppm install DateTime
ppm install DateTime-TimeZone
ppm install Template-Toolkit
ppm install Email-Send
ppm install Email-MIME
ppm install Email-MIME-Encodings
ppm install Email-MIME-Modifier
ppm install DBO-ADO
```

then run these – optional

```
ppm install Chart
ppm install Template-GD
ppm install XML-Twig
ppm install MIME-tools
ppm install PatchReader
ppm install PerlMagick
ppm install perl-ldap
ppm install Authen-SASL
ppm install RadiusPerl
ppm install SOAP-Lite
ppm install HTML-Scrubber
ppm install Email-MIME-Attachment-Stripper
ppm install Email-Reply
ppm install TheSchwartz
ppm install Daemon-Generic
ppm install mod_perl
```

To Use SQL2008

Install\enable FULL TEXT INDEXING and start service (SQL Full-text Filter Daemon Launcher(instance name)
Enable CLR integration

To use SQL2005

Enable CLR integration
Install\enable FULL TEXT INDEXING and start service (SQL Server FullText Search(instance name))

Copyright 2010 Enterim LLC Paul M Drumm

In the 487439 patch change the following.

```
+    'mssql' => {db => 'Bugzilla::DB::Mssql', db_version => '9.00.000',
-    'mssql' => {db => 'Bugzilla::DB::Mssql', db_version => '10.00.000',
```

Created a bugs database with a bugs user and pw bugs

changed localconfig in the Bugzilla directory for Mssql and the bugs user

Applied patches in order – some are complete replacements others are search and replace

used patch from https://bugzilla.mozilla.org/show_bug.cgi?id=285122
earliest patch
285122-mssql.1-2005-06-13.txt
used patch from https://bugzilla.mozilla.org/show_bug.cgi?id=487437
patch 5 for Bugzilla\DB\Mssql.pm - replacement
487437-patchv5-2009-07-24.txt
used patch from https://bugzilla.mozilla.org/show_bug.cgi?id=487438
patch 4 for Bugzilla\DB\Schema\Mssql.pm - replacement
487438-patchv4-2009-07-24.txt
used patch from https://bugzilla.mozilla.org/show_bug.cgi?id=487439
Bugzilla/Constants.pm – mssql db driver addition
487439-patchToConstants-2009-04-14.txt
used patch from https://bugzilla.mozilla.org/show_bug.cgi?id=487443
patch 2Bugzilla/User.pm
487443-patchv2-2009-04-30A.txt
used patch from https://bugzilla.mozilla.org/show_bug.cgi?id=504411
patches token.cgi, token.pm, user.pm
504411-patchv1-2009-07-22.txt
used patch from https://bugzilla.mozilla.org/show_bug.cgi?id=505362
Bugzilla/DB/Schema.pm
505362-patchv1-2009-07-24.txt

Changed $offset1 to $offset in db\mssql.pm subadjust_statement (line 563 of 487437)

-$after_where = " $random_name.R BETWEEN $**offset1** AND $limit$offset";
+$after_where = " $random_name.R BETWEEN $**offset** AND $limit$offset";

In the \db\schema\mssql.pm sub _get_create_index_ddl you have to comment out the following line in **RED** as SQL 2005 does not have filtered indexes.

Copyright 2010 Enterim LLC Paul M Drumm

Make it a filtered index IF we allow nulls certain conditions
In the case of multiple fields NOT NULL only allies when all fields are null else standard rules apply.
#$sql .= "WHERE [".join('] IS NOT NULL AND [', @$index_fields)."] IS NOT NULL\n" if $allownull;

Added quotes to ts_job_coalesce_idx => [qw("coalesce" funcid)], in DB\schema.pm (line1447 - `ts_job =>`**) Note: may be able to use \[and \] instead of quote.**

```
-ts_job_coalesce_idx  => [qw(coalesce funcid)],
+ts_job_coalesce_idx  => [qw("coalesce" funcid)],
```

Changed the default from "" to "blank" for resolution in db.pm on line 73

```
-   resolution   => ["","FIXED","INVALID","WONTFIX",
"DUPLICATE","WORKSFORME",
                "MOVED"],

 +  resolution   => ["Blank","FIXED","INVALID","WONTFIX",
"DUPLICATE","WORKSFORME",
                "MOVED"],
```

Changes to \db\mssql.pm
sub bz_setup_database –
1. add the \+ between '00' and CAST to concatenate the strings and needed to be escaped

```
            $self->do("CREATE FUNCTION DATE_FORMAT( \@date datetime,
\@format nvarchar(max) )
   RETURNS varchar(max) AS
   BEGIN
      SET \@format = Replace(\@format, '%y', DATEPART(yy,\@date))
      SET \@format = Replace(\@format, '%m',
RIGHT('00'\+CAST(DATEPART(mm,\@date) as NVARCHAR(2)),2))
      SET \@format = Replace(\@format, '%d',
RIGHT('00'\+CAST(DATEPART(dd,\@date) as NVARCHAR(2)),2))
      SET \@format = Replace(\@format, '%h',
RIGHT('00'\+CAST(DATEPART(hh,\@date) as NVARCHAR(2)),2))
      SET \@format = Replace(\@format, '%i',
RIGHT('00'\+CAST(DATEPART(mi,\@date) as NVARCHAR(2)),2))
```

Copyright 2010 Enterim LLC Paul M Drumm

```
      SET \@format = Replace(\@format, '%s',
RIGHT('00'\+CAST(DATEPART(ss,\@date) as NVARCHAR(2)),2))
      RETURN \@format
   END");
```

2. added \+ for the same reason on the function

```
   $self->do("CREATE FUNCTION REMOVE_CONTROL_CHARS( \@value
nvarchar(max))
   RETURNS bit AS
   BEGIN
      DECLARE \@I int
   SET \@I = 0;
   WHILE \@I < 32
      BEGIN
         IF (CHARINDEX(char(\@I),\@value) > 0) BEGIN
         RETURN 1
      END
      SET \@I = \@I  \+  1
   END
   RETURN 0;
   END");
```

End Changes to Changes to \db\mssql.pm

Changes to \db\schema\mssql.pm sub _get_create_index_ddl

Note: In 2008 the PK names are the first 8 of the table. Cannot have duplicate FT indexes. Need new code to have one FT index with multiple columns. BUGFIX needed.

Added about line 171

```
+ my $PKeyName="PK__".substr($table_name,0,8);
```

Changed

```
-       $sql .= "select \@PK_Index = name from sysindexes where
id=object_id('$table_name') and name LIKE 'PK__$table_name%'\n";

 +      $sql .= "select \@PK_Index = name from sysindexes where
id=object_id('$table_name') and name LIKE '$PKeyName%'\n";
```

Bad code in same module. Additions in RED

```
-   #FullText indexes require a normal index too, so will make one even if one exists.
```

Copyright 2010 Enterim LLC Paul M Drumm

```
   if ($index_type =~ /FULLTEXT/i){
      $sql .= "IF NOT EXISTS (select 1 from sys.fulltext_catalogs WHERE name =
'bugs') BEGIN\n";
      $sql .= "   CREATE FULLTEXT CATALOG bugs;\n";
      $sql .= "END\n";
      $sql .= "DECLARE \@PK_Index NVARCHAR(200)\n";
      $sql .= "select \@PK_Index name from sysindexes where
id=object_id('$table_name') and name LIKE 'PK__$table_name%'\n";
      $sql .= "EXEC('CREATE FULLTEXT INDEX  ON [$table_name] (".join('', ',
@$index_fields).")\n";
      $sql .= "KEY INDEX ['\@PK_Index']\n";
      $sql .= "ON bugs')";
```

Change to

```
+   #FullText indexes require a normal index too, so will make one even if one exists.
   if ($index_type =~ /FULLTEXT/i){
      $sql .= "IF NOT EXISTS (select 1 from sys.fulltext_catalogs WHERE name =
'bugs') BEGIN\n";
      $sql .= "   CREATE FULLTEXT CATALOG bugs;\n";
      $sql .= "END\n";
      $sql .= "DECLARE \@PK_Index NVARCHAR(200)\n";
      $sql .= "select \@PK_Index = name from sysindexes where
id=object_id('$table_name') and name LIKE 'PK__$table_name%'\n";
      $sql .= "EXEC('CREATE FULLTEXT INDEX  ON [$table_name] (".join('', ',
@$index_fields).")\n";
      $sql .= "KEY INDEX [' \+ \@PK_Index \+ ']\n";
      $sql .= "ON bugs')";
```

BUG NOT FIXED – something is creating duplicate FT indexes on the same table using a different column. SQL only allows one FT index per table.

Added new code to inhibit creation of multiple FT indexes on same table.

$sql .= "IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE NAME='$table_name' AND ftcatid>0)"; This is shown below at approximately line 192.

```
#FullText indexes require a normal index too, so will make one even if
one exists.
    if ($index_type =~ /FULLTEXT/i){
        $sql .= "IF NOT EXISTS (select 1 from sys.fulltext_catalogs
WHERE name = 'bugs') BEGIN\n";
        $sql .= " CREATE FULLTEXT CATALOG bugs;\n";
        $sql .= "END\n";
        $sql .= "DECLARE \@PK_Index NVARCHAR(200)\n";
        $sql .= "select \@PK_Index = name from sysindexes where
id=object_id('$table_name') and name LIKE '$PKeyName%'\n";
```

Copyright 2010 Enterim LLC Paul M Drumm

```
+     $sql .= "IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE
NAME='$table_name'  AND ftcatid>0)";
        $sql .= "EXEC('CREATE FULLTEXT INDEX  ON [$table_name]
(["].join("], [", @$index_fields)."])\n";
        $sql .= "KEY INDEX [' \+ \@PK_Index \+ ']\n";
        $sql .= "ON bugs')";
    } else {
        $sql .= "CREATE ";
        $sql .= "$index_type " if $index_type eq 'UNIQUE';
        $sql .= "INDEX [$index_name] ON [$table_name] (["].join("], ["],
@$index_fields)."])\n";
```

Added the following to get around the use of the keyword coalesce I had to put in quotes and followed it through. Note: maybe redundant if I used [] instead of double quotes.

if ($field =~ "\"coalesce\""){$field="coalesce";}

```
  # If all fields are NOT NULL then the index should be a standard
index
    # If it is mixed then ignore nulls
    my $allownull;
    foreach my $field (@$index_fields) {
+     if ( $field =~ "\"coalesce\""){$field="coalesce";}
```

END Changes to \db\schema\mssql.pm sub_get_create_index_ddl

Wrote CLR regexp since it was not created. Made CLR regexp function and installed in database. Use VS IDE with DB project and SQL item and publish it to DB.

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Data.SqlTypes;
using System.Text.RegularExpressions;
using Microsoft.SqlServer.Server;

public partial class UserDefinedFunctions
{
    [Microsoft.SqlServer.Server.SqlFunction(IsDeterministic = true,
IsPrecise = true)]
    public static bool REGEXP(string source, string regexp, bool
ignorecase)
    {
        if (ignorecase)
        {
            Regex r1 = new
Regex(regexp.TrimEnd(null),RegexOptions.IgnoreCase);
            return r1.Match(source.TrimEnd(null)).Success;
        }
```

Copyright 2010 Enterim LLC Paul M Drumm

```
        else
        return Regex.Match(source.TrimEnd(null), regexp.TrimEnd(null),
RegexOptions.Compiled).Success;
    }
};
```

Post Install Bug fixes

Bugzilla\DB\Mssql.pm ~ line 131 – fix in RED

```
sub sql_fulltext_search {


    #The MATCH and AGAINST functions are not in MS T-SQL and are
    #not in the where predicate so we have to use the custom CLR
    #dbo.REGEXP function instead

    my ($self, $column, $text) = @_;

    # Add the boolean mode modifier if the search string contains
    # boolean operators.
    my $mode = ($text =~ /[\-<>()~*"]/ ? "IN BOOLEAN MODE" : "");

    # quote the text for use in the MATCH AGAINST expression
    $text = $self->quote($text);

    # untaint the text, since it's safe to use now that we've quoted it
    trick_taint($text);

    #return "MATCH($column) AGAINST($text $mode)";
     return "CAST(dbo.REGEXP($column,$text,1) as int)";
}

======================AFTER INSTALL ITEMS======================
```

Configure Aliasing

There is a unique constraint on the alias field in the bugs table. If you do not want to have aliases for bugs (default) then remove this. If you want to have aliases then turn on bug aliasing by going to administration, parameters, bug fields and turn on use bug aliases. If it errors out then give the proper account access to the ..\bugzilla\bugzilla-3.4.2\data\params file. This will be either the Network Service (on a server) or the local anonymous account (IUSR_machineName) on the workstation. As an alternative you can go into the params file directly and modify the line 'usebugaliases' => '0' to 'usebugaliases' => '1'. After you turn this switch on you will see an “Alias” field on the bug form.

To remove constraint

```
DROP INDEX [bugs_alias_idx] ON [dbo].[bugs] WITH ( ONLINE = OFF )
```

To add constraint back

```
CREATE UNIQUE NONCLUSTERED INDEX [bugs_alias_idx] ON [dbo].[bugs]
(
```

Copyright 2010 Enterim LLC Paul M Drumm

```
        [alias] ASC
)
```

Copyright 2010 Enterim LLC Paul M Drumm

==========Use of Active Perl's PerlIS.dll==========

If you are going to use the perlis.dll instead of the perl.exe you need to do the following.
Note: if you put all the cgi files into a Visual Studio solution then you can do this globally.

Remove the taint check ("T") from the top of all the cgi pages.

Change

```
#!/usr/bin/perl –wT
```

To

```
#!/usr/bin/perl -w
```

If you do not like seeing the header printed at the top of the pages while using the perlis.dll you can do the following on all the cgi pages.

Find

```
# Return the appropriate HTTP response headers.
print $cgi->header();
```

and comment out the second line

```
# Return the appropriate HTTP response headers.
#print $cgi->header();
```

Find this line on enter_bug, show_bug, page, and query cgi pages

```
print $cgi->header($format->{'ctype'});
```

and comment it out

```
#print $cgi->header($format->{'ctype'});
```

In the internet server administration MMC, go into the application configuration and change the mappings for .pl, .plx, and .cgi to use the perlis.dll typically found in the C:\perl\bin directory.