

SABIC UK Petrochemicals/Teesside University

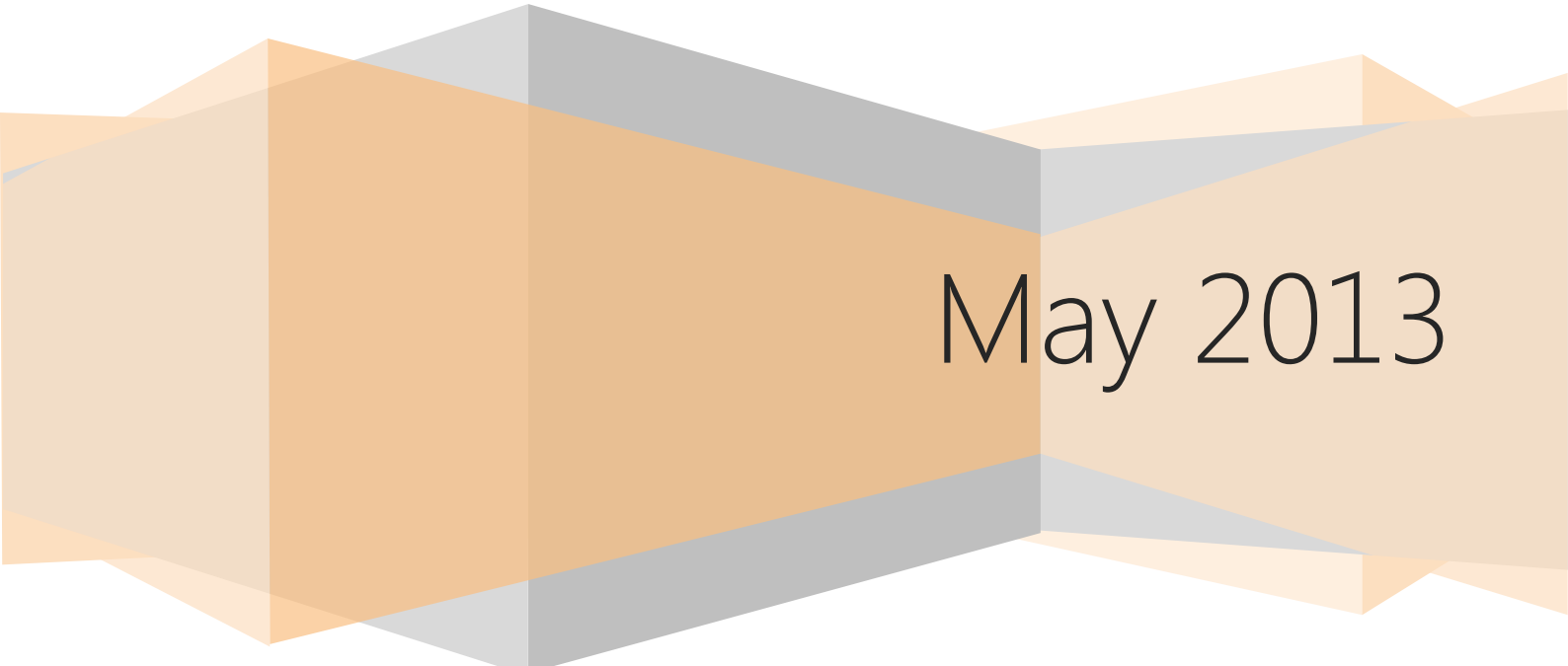


Teesside  
University

# Student Placement Report

Stephen Pammenter

Applications Developer

An abstract graphic at the bottom of the page consists of several overlapping, semi-transparent geometric shapes in shades of orange and grey, creating a 3D effect.

May 2013

## PREFACE

It is assumed that the reader of this report is from a technical background, and understands the terminology and syntax used throughout, without additional explanation required from the author. Code used in this document is represented using the `Courier New` font, and will be coloured in the same style as would be found in Microsoft Management Studio and Microsoft Visual Studio.

All examples of code used throughout this report is property of SABIC UK Petrochemicals<sup>1</sup>. Some screen captures in this document contain sensitive information, and have been blurred to protect company data.

---

<sup>1</sup> Registered Number: 03767075, Registered Office: The Wilton Centre, Redcar, Cleveland TS10 4RF.

## CONTENTS

PREFACE .....	II
CONTENTS.....	III
1. INTRODUCTION.....	4
2. MAIN SYSTEMS WORKED ON.....	5
2.1 Site Plan Tracker.....	5
2.2 EU-ETS Data Extraction.....	6
2.3 EU-ETS KPI's .....	9
3. CONCLUSION.....	10
4. APPENDIX .....	I
Item 1. Site Plan Tracker – Charter.js .....	I
Item 2. EU-ETS Extraction.....	I
Item 3. EU-ETS Extraction Agent.....	III
Item 4. EU-ETS KPI – SIMS Events Widget.....	IV

## 1. INTRODUCTION

The previous seven months have marked a milestone in my time here at SABIC. And I feel that this period has seen me develop from student, to professional where I have fully integrated into my role as an Applications Developer.

As my experience has increased, so has my work-load; where there have been several occasions I have been required to meet tight deadlines, and still deliver a quality product. I have performed a greater amount of service to customers such as in the conclusion to the Site Plan Tracker project. Each challenge that has come my way I have accepted and dealt with accordingly and once again have thoroughly enjoyed every step of the way.

## 2. MAIN SYSTEMS WORKED ON

I have performed less work on web applications in this report than I have done previously, with much of my work focus being directed towards an extraction agent. This is not to say that the work I have done has not involved web development. The work listed below has been in alignment with major Teesside projects, and work like my extraction agent has been business critical.

### 2.1 SITE PLAN TRACKER

The follow on from my previous report on Site Plan tracker has been two fold. Firstly in February the system advanced from general availability to full release on the Teesside site. This came in the form of three major launch events to senior and team managers across the site, and required me to be on hand to assist with setup and provide demonstrations.

Second was feedback from the events which was to make adjustments to usability of the system. I am currently working on an overhaul to the UI of the system, improving its usability and styling it in line with the SABIC theme. As can be seen in **Error! Reference source not found.** below

The figure displays two versions of the 'Evolution Plan - Part 1 Charter' form. The left version is the older UI, characterized by a light blue background and a more cluttered layout. It includes sections for 'Evolution Theme Alignment', 'Area of Responsibility', 'Improvement Definition', and 'Business Impact'. The right version is the newer UI, featuring a dark blue background and a more modern, fluid layout. It also includes the same sections but with a more organized structure and a 'Charter Status: New' indicator at the top right. Both forms include fields for 'Title of Charter', 'Sponsor', 'Delivery Manager', 'Team Leader', 'Delivery Method', 'Term', 'Charter Priority', 'Delivery Difficulty', and 'Opportunity Value'. The new UI also includes a 'Business Impact' section with a 'This field is mandatory' note.

Figure 1 - Site Improvement Tracker UI Changes

The new version (on the right) features a fully fluid layout that works on all resolutions (a problem we highlighted at our launch events); another major change has been the removal of page re-directs between screens. The entire charter is now loaded to the page, and screen changes are done through JavaScript, providing a fast seamless user experience.

I stripped out the existing CSS, and replaced it with a much better layout. It is now easily readable; moreover, it is easy to maintain. To make the page fluid I use only percentages - and apply a minimum width where necessary. I have also made full use of the '*em*' unit, a unit used to proportion size to its parent. This has allowed me to make use of the JavaScript seen in **Item 1**. By changing the body font size, the browser will automatically re-render sizes for the rest of the page.

I have enjoyed making these changes as I have learnt a great deal about modern front-end techniques, and a great deal about CSS3 and HTML5. Although the task has proven to be large due to the static nature the ASPX form it was built on, I hope to see it to the end.

## 2.2 EU-ETS DATA EXTRACTION

To provide some context to the work I have been doing I will first explain what EU-ETS is and the impact it is having on SABIC in Teesside. EU-ETS is the European Union - Energy Trading Scheme which is a system set up to address climate change by applying a cap on the amount of CO<sup>2</sup> an industrial installation can make. There are several phases to EU-ETS; SABIC recently progressed to Phase III which requires us to begin reporting our emissions.

The EU-ETS project is a large site wide project, which has been running for several years. Data is continually collected from various instruments on our plants and fed into our IP21 data historian servers. My supervisor assigned most of my resource to the project and I was required to deliver an automated system that would, accurately extract and report quickly accessible EU-ETS data.

The application is scheduled to run on a daily basis, and collect data from IP21; this required me to learn SQL+ (an AspenTech variation of SQL) and learn how data was collected, calculated and stored within records in IP21.

The first build I made mainly used Excel to get the data, and the agent simply copied values across to an output workbook. I used a Sabisu add-in to import a data table into a cell array directly from our IP21 server.

We then released this first version after successful testing; the next iteration was required to further export data into calculation workbooks. The solution I provided was to grant users access to a script for them to reference a tag (the name of an IP21 value), select a workbook, location and export the data into it. I started out by writing an API that would provide the basis of the application, and then began building the functions in C# that would be required. I built an interpreter process to read the script, and call the functions required see **Item 2**.

Below is a sample of the script I used in my end-user guide:

```
//Open the sheet you are copying into.  
OPEN G:\Energy Management\EU ETS\Data  
Management\MyFile.xls  
//Select the sheet the data is going into.  
SHEETOUT Sheet1  
//Select the tag type, tag, area and cell reference data  
is going into.  
COPY Daily FI9401.PV Olefins INTO A6
```

Figure 2 - End User Extraction Script

After the functions were completed to meet the API spec, I then released the application to run on our server, under a service account. This was where we met problems which instigated the third iteration of the application. Due to the way the Sabisu add-in works – a requirement to have a recognised active directory account registered - I was unable to get the data into excel using the add-in. SQL+ is rather limited into what it can do with data manipulation, so to provide a solution to both these problems; we decided to import data to our largely accessible SQL servers. The final process looks like this:

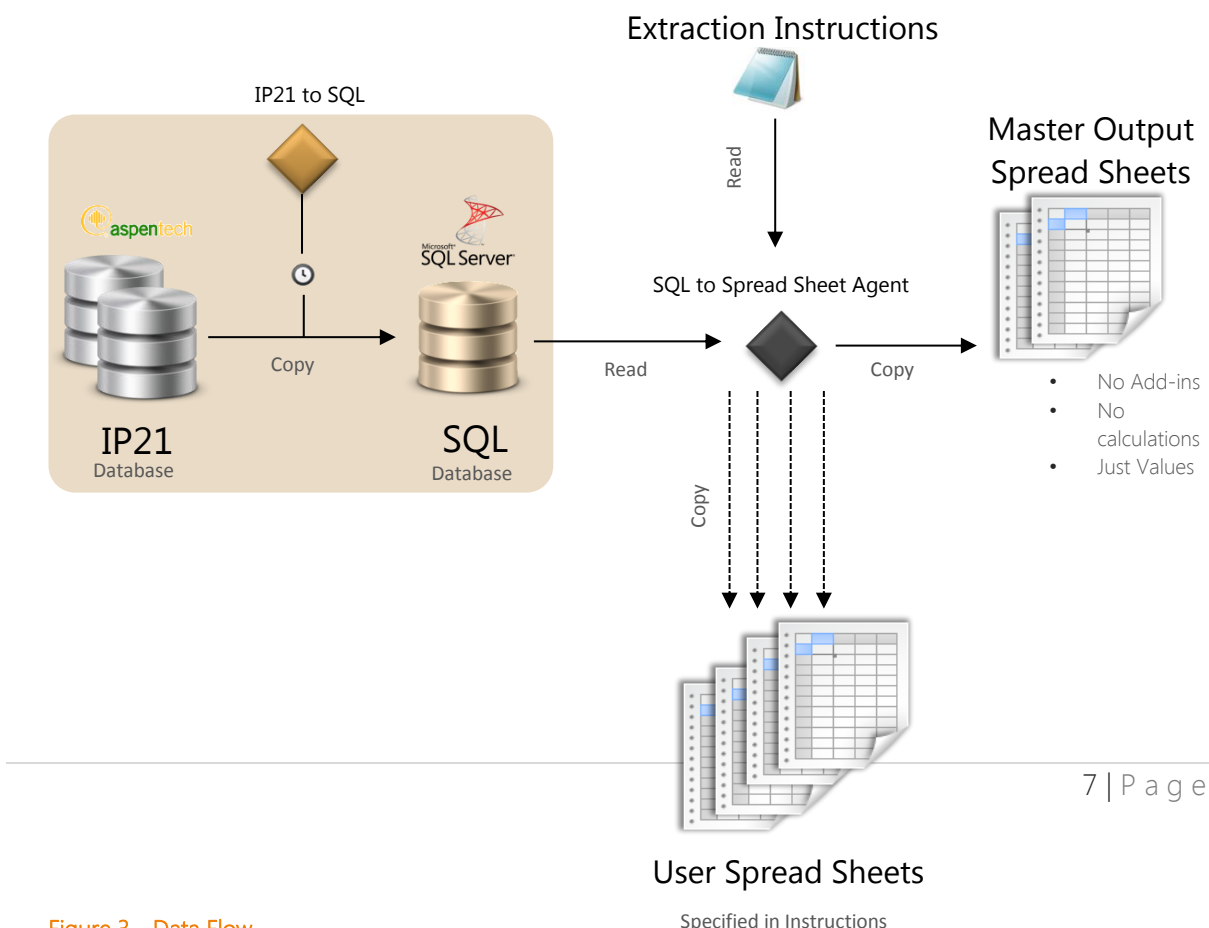


Figure 3 - Data Flow

A later addition to the application was allowing the use of year functions, which would allow the finding and creation of directories. I added %YEAR% to the script, which is converted by the application to the current year. Instead of creating a loop to copy files from one directory to another, I elected on the use of the /robocopy function of CMD, as seen in **Item 3**.

I learnt a great deal from developing this application; one of the major things was about using Microsoft's Interop library.

I also gained a deeper appreciation for memory allocation. To interact with Excel, I had to use COM objects which are renowned for being hard to work with. One of the major requirements was to ensure that the application fully ended all of its processes else they would have been trailing Excel processes running on the server, which I had major problems with at one point. To combat this issue I had to symmetrically reference, and de-reference all COM object. I followed a 'two dot rule', where I would never use more than two dots in when assigning a COM object. Such as:

```
Excel.Worksheet sheet = excelApp.Worksheets.Open("MyFile.xls");  
//Would need to be:  
Excel.Worksheets sheets = excelApp.Worksheets;  
Excel.Worksheet sheet = sheets.Open("MyFile.xls");
```

Figure 4 - 'Two Dot' Rule

I also used ReleaseCOMObject() to ensure that I fully released COM objects with:

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(Obj);
```

Figure 5 - Use of Marshal

I met all of the deadlines set of me, and as with agile methodologies I delivered a working application early on. Testing has proved the application to be accurate, and in emission reporting even a 2% error could give the company a six figure fine – a testament to the involvement of a placement student within SABIC. I am very proud of this application, and it has not only received appraisal from my supervisor, but also from members of our site leadership team.



## 2.3 EU-ETS KPI's

I am currently working on reporting key performance indicators from our EU-ETS data. This has expanded on my knowledge of SQL+ and our IP21 systems. One of my tasks was to report EU-ETS related safety incidents. To do this, I had to get a count of events and pivot the table to allow Sabisu to read data series into a graph. I have shown the SQL code in **Item 4**.

This work was made a lot easier by the earlier foresight to import IP21 data into SQL. This system is now in use and my overdue approvals widget has already highlighted some problems on the plants.

In this period I have also produced a large amount of widgets in Sabisu which is always fun to do as it combines many of the skills I have written about previously. And I feel that the styling of many of my widgets has reflected the development of my front-end knowledge.

The KPI's are currently being validated, but the combination of the above work has resulted in a concise dashboard in Sabisu:

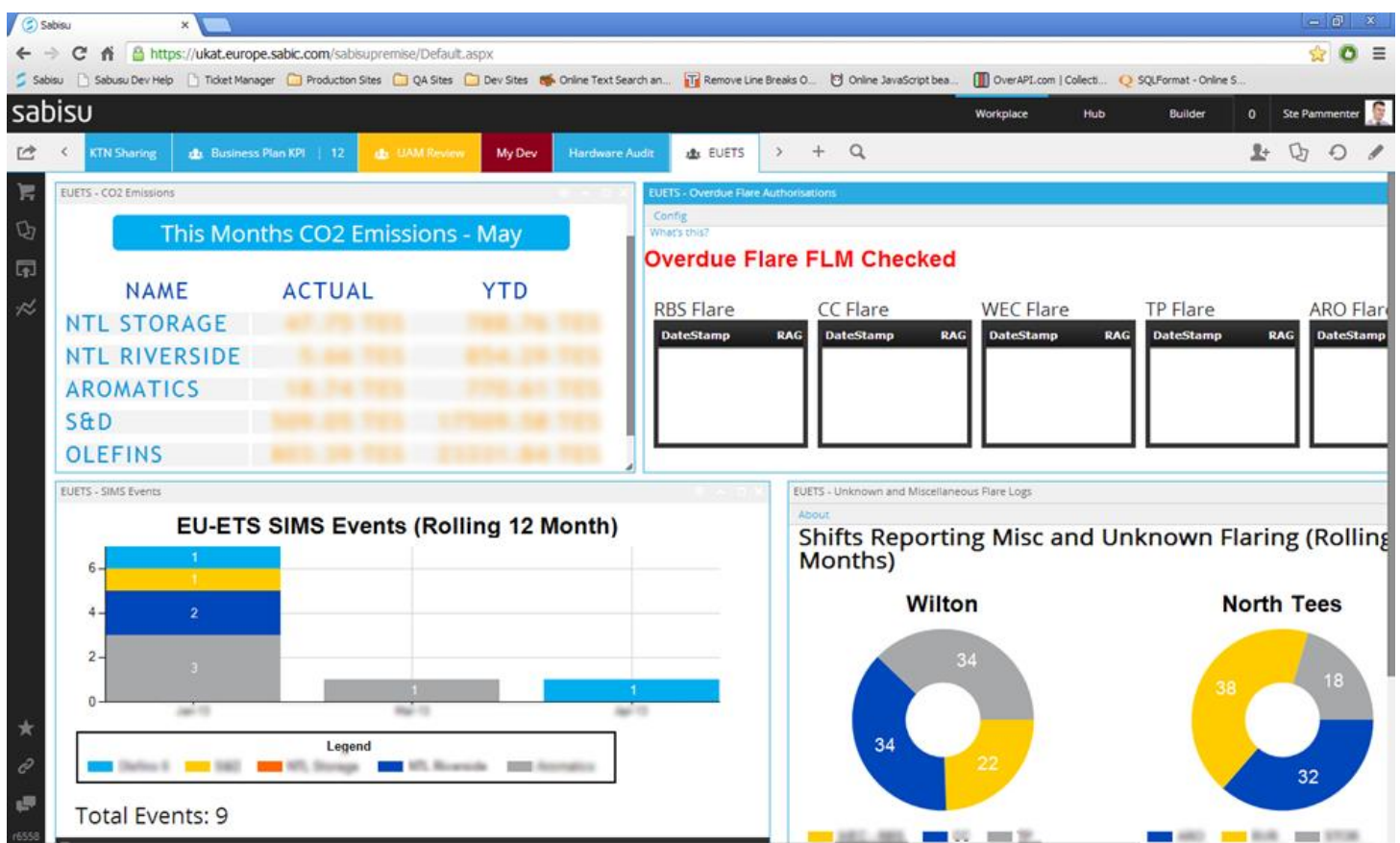


Figure 6 - EU-ETS Dashboard

### 3. CONCLUSION

Since my last report, much of the work I have done has required more communication with customers. The EU-ETS project meant communicating in a variety of different styles from large meetings to one-on-one consultations with Process Engineers. The Site Plan Tracker launch events required me to provide professional help to senior managers, where it was paramount that I represent my team well. I feel that from this variety of social scenes, my communication, intuition and team work skills have all improved.

I have also built up a greater judgement of time to spend on work, and when to tell a customer I could or could not do a task due to work I already have on. Although larger pieces of work are managed by my supervisor or PM, ticket work is very much down to the judgement of a developer. And I have learnt from my mistakes of erroneous acceptance of work, where there was no way I could get it done. I find that work scope is fundamental in this, and have found myself asking a considerable amount more questions before I even start ticket work to gain a much clearer specification.

## 4. APPENDIX

### Item 1. SITE PLAN TRACKER – CHARTER.JS

On a resize event, the `dynamicLayout()` function is called this checks the page width and adjusts the body font size accordingly. This simple light weight JavaScript then causes element sizing to filter down to the lowest child.

```
window.onresize = function onResize() {  
    dynamicLayout();  
}  
  
function dynamicLayout() {  
    var browserWidth = getBrowserWidth();  
    if (browserWidth < 740) {  
        document.body.style.fontSize = '12px';  
    }  
    else if (browserWidth > 740 && browserWidth <= 887) {  
        document.body.style.fontSize = '14px';  
    }  
    else {  
        document.body.style.fontSize = '16px';  
    }  
}
```

### Item 2. EU-ETS EXTRACTION AGENT – INTERPRET FUNCTION

Overleaf is a reduced version of my interpret code, it shows where the function reads each line from the extraction instructions and breaks up lines into words. It then operates a switch statement based on the first word – the command word, of that line.

```
[STAThread]
private static void Interpret()
{
    string[] strFileLines;
    char chDelimiter = ' ';
    string[] strWords;
    string strOneLine;
    int i = 1;

    //Default for Master to excel.
    blnMasterInput = true;
    blnIncludeFormat = false;
    blnDebug = true;
    WriteEventLog("Opening new instance of excel for Interpret.",
EventLogEntryType.Information);
    WriteToDebug("----- Agent Run :: IP21 to Excel -----",
false);
    WriteToDebug("----- " + DateTime.Now.ToString(DateFormat) + " ---
-----", false);
    blnDebug = false;
    //Check that instructions exist.
    if (System.IO.File.Exists(strInputLocation +
"AgentData\\ExtractionInstructions.txt"))
    {
        //Break text into lines.
        strFileLines = System.IO.File.ReadAllLines(strInputLocation +
"AgentData\\ExtractionInstructions.txt");
        //Read line by line.
        foreach (string strLine in strFileLines)
        {
            strOneLine = strLine;
            //Remove any additional white spaces
            while (strOneLine.IndexOf(" ") > -1)
            {
                strOneLine = strOneLine.Replace(" ", " ");
            }

            strWords = strOneLine.Split(chDelimiter);

            //First word is command, perform that commands function.
            switch (strWords[0].ToUpper())
            {
                case "//":
                    //Comment, do nothing.
                    break;
                case "":
                    //Blank line, do nothing.
                    break;
                case "DEBUG":
                    blnDebug = !blnDebug;
                    break;
                case "OPEN":
                    WriteToDebug(i, "Opening file...");
                    Close(false);
                    Open(strOneLine, false);
                    break;
                {...} //case then for others...
            }
        }
    }
}
```

### Item 3. EU-ETS EXTRACTION AGENT

Here I have taken out part of my open function in the agent to demonstrate the use of CMD to copy directories and files.

```
//Check to see if this contains a year function.
if (Location.Contains("%YEAR%") || Location.Contains("%NEXTYEAR%"))
{
    strLocationThisYear = Location.Replace("%YEAR%", dtToday.Year.ToString());
    strLocationThisYear = strLocationThisYear.Replace("%NEXTYEAR%", (dtToday.Year + 1).ToString());
    //Check to see if a directory for that year doesn't exist, inferring a new year.
    if (!Directory.Exists(Path.GetDirectoryName(strLocationThisYear)))
    {
        strLocationLastYear = Location.Replace("%YEAR%", (dtToday.Year - 1).ToString());
        strLocationLastYear = strLocationLastYear.Replace("%NEXTYEAR%", dtToday.Year.ToString());
        // /C ensures that CMD exits after it has completed its process. /E ensures that all
subfolders are copied too.
        strCMD = "/C robocopy \"" + Path.GetDirectoryName(strLocationLastYear) + "\" \"" +
Path.GetDirectoryName(strLocationThisYear) + "\" /E";
        processCMD = Process.Start("CMD.exe", strCMD);
        //Pause application until CMD is complete.
        processCMD.WaitForExit();

    }
    Location = strLocationThisYear;
}
}
```

## Item 4. EU-ETS KPI – SIMS EVENTS WIDGET

The SQL code below shows how I create a table of events by pivoting the view

dbo.SIMS\_Events.

```
SELECT
    LEFT(DATENAME(MONTH, MonthName), 3) + '-' + RIGHT(DATENAME(YEAR, MonthName), 2)
AS MonthName,
    Aromatics,
    [NTL Riverside],
    [NTL Storage],
    [S&D],
    [Olefins 6]
FROM
    --Build list of events by area.
    (
        SELECT
            AreaName,
            CAST('01-' +
master.dbo.fn_formatDateToMonth(master.dbo.fn_UKDateStringToDateTime([Date], '/')) AS
DATETIME) AS MonthName,
            [Event Id]
        FROM
            SIMS_EUETSEvents
        WHERE
            CAST('01-' +
master.dbo.fn_formatDateToMonth(master.dbo.fn_UKDateStringToDateTime([Date], '/')) AS
DATETIME) BETWEEN DATEADD(YEAR, - 1, GETDATE()) AND GETDATE() AND CAST('01-' +
master.dbo.fn_formatDateToMonth(master.dbo.fn_UKDateStringToDateTime([Date], '/')) AS
DATETIME) >= '01-JAN-2013'
    )
    --Count events and pivot the table using area's as headings.
p PIVOT (
            COUNT([Event Id]) FOR AreaName IN ([Aromatics], [NTL Riverside], [NTL
Storage], [S&D], [Olefins 6])
        ) AS Pvt
```