



# CCS Developer

NOVEMBER 2004

VOLUME NO. 1 ISSUE NO. 4

DataObjx L.L.C.

Website: [www.dataobjx.net](http://www.dataobjx.net)

DataObjx L.L.C.  
Meriden, CT USA

Office: 860/919-2536

Email: [administrator@dataobjx.net](mailto:administrator@dataobjx.net)

Website: [www.dataobjx.net](http://www.dataobjx.net)



DataObjx L.L.C.

**Write Articles For  
CCS Developer  
Magazine.**

**Building  
Community**



Need More Answers?  
Click [here](#) to access the Code  
Charge Studio Forums

## In This Issue

**Dynamic Calendar Creation - (ASP)** – We continue our Expanding the CodeCharge Studio Portal series with an article you're sure to use. In this article you will learn how to use a vbClass to generate your calendar. Simply add the SQL and set a few parameters to configure the look and feel of your calendar and you're done.

**Dynamic Calendar Creation – PHP** - Edwin Brown shows you how to generate a dynamic calendar using PHP. If you're attempting to build a calendar for your applications, these are the functions and techniques you need to know.

**RSS 2.0 Feed Generator Functions - PHP** - Femi Hasani shows you how to create an XML Feed for your web site. RSS is being used more often than ever before, so if you're considering creating BLOGs or simply want others to know more about your site, this article demonstrates the techniques you need to know.

**SQL To Excel – ASP** – We go to all that effort to create database driven applications and what does the end user want? An Excel Spreadsheet containing their data! Or, maybe you just want a back-up of a table before you execute that 'tricky' set of routines you just created. This article shows you how to generate an Excel Spreadsheet containing data – based on a SQL Query.

# Editors Comments

## Viewpoint...

We hope you find this articles in this issue beneficial. We think you will.

In this issue, we feature two articles from two new contributors to CCS Developer Magazine. We thank Mr. Femi Hasani of [www.Vision.To](http://www.Vision.To) for his article on RSS Feeds and Mr. Edwin Brown of <http://www.mambobrown.com> for his article on dynamic calendar creation using PHP.

This months issue will be important for CodeCharge Studio developers using .ASP or .PHP who want to incorporate dynamic calendar capability into their applications.

Both articles demonstrate different approaches and methodologies to the same issues and one will certainly benefit by reading both articles even if you don't program in the other language.

Each calendar can be customized or modified to fit your particular requirements are both are easy to integrate into your code charge studio applications.

If you've ever considered adding an RSS Feed to your web site, Mr. Hasani will show you how.

RSS Technology is important for those who want to create BLOG applications, or derive RSS feeds from other sites. This article will get you started.

In addition, we've added an article that demonstrates a technique you'll find useful in virtually all of your applications – Generating Excel Spreadsheets based on a SQL query.

["Hmm, if only there was some place I could sell my code"](#)

There is!

We launched the CCS MarketPlace this week and we're ready to start populating it with project files and/or solutions that you have and want to sell!

For those of you who have .ASP capable websites and want to get your CodeCharge Studio Store project on-line, we've encapsulated the logic and routines necessary to let your cart communicate with PayPal payments and the PayPal Instant Payment Notification System.

We can't stress how important it is that you consider writing an article for CCS Developer Magazine. Articles from people like Fernando, Sixto, Edwin & Femi have given many of you the ability to create applications you couldn't create before. We therefore ask that you consider what contribution you can make to the community and what that will mean – to the developer like you – who seeks a solution to their programming problems.

Please contact [administrator@dataobjx.net](mailto:administrator@dataobjx.net) and let us know that you have an idea for an article. We'll publish it so that others can benefit from your knowledge or special routines.

In addition, we really need to hear from you regarding the types of solutions you'd like to see in CCS Developer Magazine!

We don't want to publish articles on things that don't interest you. What would be the point?

We need you to send us your idea's right away. It's a lot of work choreographing a magazine, believe me – so we need to know your 'needs' or the magazine will not be helpful.

Our greatest intention is that you derive more value from one or two of the solutions (in terms of shortened development time and learning time) than the cost of the magazine. So it's up to you to maximize the value you derive and the administrator needs to hear your voice.

The solutions provided in this issue are far ranging and can be applied in a myriad of ways to achieve different outcomes.

One more thing.

We try to ensure that the code that is delivered with the magazine functions as expected. We're getting better at managing this process but the unexpected sometimes happens and a file is missing or an image file is missing. We apologize for this and ask that when this happens you let us know so that we can send out updates. The same goes for reporting bugs.

Best Regards,

Martin Hamilton  
[www.dataobjx.net](http://www.dataobjx.net)

# Expanding The CodeCharge Studio Portal – Part 3

## Dynamic Calendar Creation With CCS.

### Events

The CodeCharge Studio portal is a great start for anyone attempting to set up an intranet or portal style web site.

Contained within the data schema is the “Events” table. To implement this solution, you’re going to need to add a few additional fields to the Events table.

In fact, once you’ve gained a complete understanding of the code, you’re likely to add even more capabilities to your dynamic calendar, adding more fields as you go.

You will also add some additional CSS (Cascading Style Sheet) to determine how each ‘type’ of event is displayed.

### The Calendar “Class”

Microsoft extended the capabilities of IIS to understand “Class Modules”. There is a lot of information on the Internet about building classes, so we’re not going to address that here.

But essentially, you ‘define’ a class by creating a certain code structure. That code structure is recognized/interpreted as a COM object and is called in the same way as a COM object.

If you have incorporated SMTP or CDOSys or CDONTS, you are already familiar with making ‘calls’ to COM objects – even if you didn’t realize you were doing so.

COM objects have properties, methods and events.

If we had a COM object called “Dog”, we might make calls to the COM object like this;

```
Dim objDog
```

```
Set objDog = New clsDOG
objDog.Legs = 4 (Property)
objDog.Tails = 1 (Property)
objDog..WagTail (Method)
objDog.Walk (Method)
objDog.On_Walk (Event)
```

It will only take a few minutes and to incorporate the dynamic calendar. Afterwards you’ll no doubt want to add what you’ve learned to your own applications.

In this article, we’re going to use a class to make our job easier and make it a simple matter to transport and incorporate this code in any of your projects.

Before we begin to explain the calendar class, we’d like to extend our thanks to **Author: Jacob Gilley** for the initial Calendar Class module.

DataObjx has extensively modified this code to incorporate additional capabilities and you are encouraged to add any additional capabilities required in your applications.

### Let’s Look Under The Hood

Open the Calendar Project and then open the Calendar.ccp/.asp file. Navigate to the Calendar\_events.asp page and scroll down to around line 219. The Calendar class begins there.

The original class code has been included along with the source code for this article.

Looking at the class you will notice the various properties that are defined as well as a number of functions (methods) that are used by the class.

We’ll only deal with the most important ones, but it won’t take you long to begin adding your own requirements into the project.

You can study the calendar class at your leisure, but for now we’re going to take a look at the changes we need to make to the “Events” table and then how to implement a dynamic calendar for your site.

### Creating / Modifying The Event\_Categories Table

The MS Access databases that deploy

with CodeCharge Studio project simply has a categories table that is used for the ‘articles’ table and you may have been tempted to use it for the events table as well.

But we encourage you to create a new table called ‘event\_categories’ based on the categories table, and to add another field to that table called ‘css\_style’ as varchar(75).

In this field, you will place the name of the CSS style that you want this type of event to display with.

The calendar class will use this field to alter the look of each ‘category’ of event automatically.

Note: You must create this table in order to make this example work. Refer to Figure #4 for the Data Schema for this table.

### Modifying The Events Table

We’ve taken the original “Events” table and added a number of additional fields.

Refer to Figure 1 for the changes that were made.

You don’t need to make these modifications to your events table. We created them specifically for the example to demonstrate that you only need to change the names of the date fields and table name in the code to get the right recordset.

Whether you include these fields in your events table or not is up to you. But some of them, such as the start\_time and end\_time fields, may be useful in certain situations, such as when reserving corporate board rooms for meetings, for instance.

The important fields remain the date\_start and date\_end fields as they are the fields that determine whether the event will appear on the given month/page of the calendar.

*Note: These fields may be named differently in your Events Table.*

### Tricky SQL

## Expanding The CodeCharge Studio Portal – Part 2

Dynamic Calendar Creation With CCS, continued...

Language: .ASP  
Database: Any

The tricky part about getting the calendar to work properly is getting the SQL right.

Performing a simple "Select \* From Events Where date\_start >= '11/01/2004' and date\_end <= '11/28/2004'" simply isn't going to work.

Why?

Because this type of SQL fails to pick up 'Long Term' events – or – events spanning a number of months.

Using the example provided above, imagine that an event begins on October 1, 2004 and ends on December 31, 2004 – Perhaps it's a Sale Special.

In this situation, the SQL above will not retrieve the "Sale Special" record and so would not display anything relating to this event on the calendar if the user was viewing "November".

Instead, we need to use the SQL "BETWEEN" command and we have to perform a couple of "Or" conditions in the process.

You can refer to Figure 2 or look at the code near line # 96.

The important thing to note about this SQL is the Where clause that is used.

Notice how we tell the database to return any records where *today's date* is BETWEEN two dates; date\_start and date\_end respectively.

By giving the database this command, we will derive the "Sale Special" record from our example above.

Since we've provided the class and SQL required to build the calendar - the rest is easy. So, let's take a look at how one goes about putting all of this together in a nice simple 'package' that can be transported easily from project to project.

[Incorporating The Class Module](#)

**Figure 1 - MODIFIED EVENTS TABLE (SQL SERVER 2000)**

```
CREATE TABLE [events] (
    [event_id] [int] IDENTITY (1, 1) NOT NULL ,
    [category_id] [int] NULL ,
    [is_approve] [bit] NULL CONSTRAINT [DF_events_is_approve] DEFAULT (0),
    [user_id_approve_by] [int] NULL ,
    [date_add] [smalldatetime] NULL ,
    [date_start] [smalldatetime] NULL ,
    [date_end] [smalldatetime] NULL ,
    [event_name] [varchar] (250) NULL ,
    [event_desc] [ntext] NULL ,
    [presenter] [nvarchar] (100) NULL ,
    [location] [ntext] NULL ,
    [location_url] [nvarchar] (255) NULL ,
    [image_url] [nvarchar] (255) NULL ,
    [fee] [nvarchar] (150) NULL ,
    [user_id_add] [int] NULL ,
    [attendees] [text] NULL ,
    [start_time] [varchar] (50) NULL ,
    [end_time] [varchar] (50) NULL ,
    [approved_date] [datetime] NULL ,
    [EmailNotificationSent] [bit] NULL CONSTRAINT
[DF_events_EmailNotificationSent] DEFAULT (0),
    [dept_id] [int] NULL ,
    [user_id] [int] NULL ,
    [web_site_url] [varchar] (150) NULL ,
    [is_public_holiday] [bit] NULL CONSTRAINT [DF_events_is_public_holiday]
DEFAULT (0),
    CONSTRAINT [PK_events] PRIMARY KEY CLUSTERED
    (
        [event_id]
    ) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Your going to add a new page to this project and incorporate the code for the dynamic calendar into your 'test page'.

It will only take a few minutes to incorporate the dynamic calendar and afterwards you'll no doubt want to add what you've learned to your own applications.

But before you 'jump the gun' and move your mouse to fire up your own project, finish this tutorial first because you'll quickly see why we added some of the fields that we did to the Events table. As a consequence, you may decide to add some of them to your events table before you incorporate the dynamic calendar in your application.

[Providing A Framework For The Calendar](#)

You may need to have a calendar that

displays events for a single user, a department or the entire corporation. In addition, you may have used the "Task Manager" solution provided by CodeCharge Studio and realized that a dynamic calendar would be beneficial.

However, you need to provide a framework within which the calendar can be presented or navigated by the user beyond traversing the month to month views.

The user also may need to view various layers of events. For example, there could be Corporate Events, Department Events and Personal Events.

Using the technique we outlined in Issue #3 – "Building Navigational Menus" we're going to build a menu for our calendar that allows the user

# Expanding The CodeCharge Studio Portal – Part 3

Language: .ASP  
Database: Any

Dynamic Calendar Creation With CCS, continued...

**Figure 2 – CORE SQL USED TO DERIVE EVENTS FROM THE EVENTS TABLE (SQL SERVER 2000)**

```
sSQL = "SELECT * FROM Events "
sSQL = sSQL & "WHERE "
sSQL = sSQL & "(" & dteToday & " BETWEEN date_start AND date_end) OR "
sSQL = sSQL & "(" & dteStart & " BETWEEN date_start AND date_end) OR "
sSQL = sSQL & "(" & dteEnd & " BETWEEN date_start AND date_end) OR "
sSQL = sSQL & "(date_start BETWEEN " & dteStart & " AND " & dteEnd & ") OR "
sSQL = sSQL & "(date_end BETWEEN " & dteStart & " AND " & dteEnd & ")"
```

View this calendar on-line at: <http://www.dataobix.net/dynamiccalendar> and navigate to Oct 2004 – Dec 2004

to traverse All Events, Corporate Events, Department Events and Personal Events.

Naturally, we'll need to modify our SQL accordingly, but first – the menu.

Open the calendar\_menu.ccp file. Look at the events page. If you purchased Issue #3 – this code will be very familiar to you.

The only important aspect about the menu – other than the technique used to produce it, is the querystring value "view".

By altering the "view" passed to the page via the querystring (URL), we can cause calendar.asp to execute the appropriate SQL necessary to produce the calendar's event data.

In this case, your possible views are;

- All Events,
- Corporate Events,
- Department Events, and
- Personal Events

In Figure #2 you can view the "Core"

SQL used in the database query, but we need to extend that by providing a wrapper of logic around it.

This "wrapper" will guide the concatenation process of the SQL query to produce the results we want.

Open the calendar.ccp events page and scroll down to about line 90. This is where you begin to add additional "WHERE" clauses to the SQL according to the "view" passed to the page.

*IMPORTANT: The other thing to notice is that a view of "corporate" is the default in this example. Keep this in mind when you configure the "default view" for your calendar.*

After the SQL has been created, the rest is pretty easy since the vbScript Class takes over the processing and 'drawing' of the calendar.

You probably want to begin using the calendar right away, so let's take a quick look at the various "Properties" that you need to "set" to make the Class function properly.

Refer to the code briefly shown in Figure Three (3) and also located within the Function lblCalendar\_BeforeShow() in the calendar.ccp events page.

This code works with the calendar class located at the bottom of the page.

After you gain an understanding of what the calendar class is doing and how it does it, you may decide to add additional properties of your own.

### Setting Up A New Calendar Page

The important thing to remember though, is this "Core Calendar Class" will need to be added to the bottom of your

"yourcalendarpagename\_events.asp" page. You simply need to copy this class and paste it into any new page you decide to use to create a dynamic calendar.

For instance, if you decided that having a dynamic calendar for the CodeCharge Studio Task Manager Application would be beneficial, you would create a page, called for instance; "TaskCalendar.ccp".

Open your TaskCalendar Page and using the "Grid Builder" place a grid on the page.

Set the DataSource of the grid to the table from which you'll derive you calendar data. In this case, from the "Tasks" table. Name the grid "Tasks".

In our example however, we are using the "Events" table. Using the "Tasks" table is simple another example.

This grid will not actually be visible.

**Figure 3 - labelname\_BeforeShow()**

```
' Instantiate A New Instance Of The Calendar
Set MyCalendar = New Calendar

MyCalendar.TableStyle = "LightWalkFormTABLE"
MyCalendar.TitlebarColor = "LightWalkColumnTD" 'Style for titlebar
MyCalendar.TitleBarDayColor = "LightWalkFieldCaptionTD"
MyCalendar.TitlebarNavigatorLink = "LightWalkNavigatorLink"
MyCalendar.CellDayColor = "LightWalkDataTD"
MyCalendar.ActivityColor = "EventActivityDataTD"
MyCalendar.CellCurrentDayColor = "EventTodayTD"
MyCalendar.HyperlinkStyle = "EventDataLink"
MyCalendar.CalendarButton = "EventButton"
MyCalendar.CalendarSelect = "EventSelect"
MyCalendar.DayViewPage = "EventView.asp"
```

# Expanding The CodeCharge Studio Portal – Part 3

Language: .ASP  
Database: Any

## Dynamic Calendar Creation With CCS, continued...

**Figure 4 – Event\_Categories Table Data Schema**

```
CREATE TABLE [event_categories] (
    [category_id] [int] IDENTITY (1, 1) NOT NULL ,
    [category_name] [nvarchar] (50) NULL ,
    [added_by_id] [int] NULL ,
    [date_added] [smalldatetime] NULL ,
    [approved] [bit] NULL ,
    [approved_by_id] [int] NULL ,
    [date_approved] [smalldatetime] NULL ,
    [css_class] [varchar] (75) NULL ,
    CONSTRAINT [PK_event_categories] PRIMARY KEY CLUSTERED
    (
        [category_id]
    ) ON [PRIMARY]
) ON [PRIMARY]
GO
```

**IMPORTANT NOTE**

Since the SQL used in this example requires this field, you need to ensure that you have created the event\_categories table shown in Figure Four (4).

To: Set cn=New  
clsDB\_YOURCONNECTION

**In Closing**

Let's summarize.

Add the Event\_Categories Table to your database.

It is placed on the page simply to cause CodeCharge Studio to provide us with the datasource that we need for our Custom Query.

Therefore, after setting the datasource of the grid to the "Tasks" table, you need to add code to the BeforeShow() event of the grid. You don't want the grid to be visible, so you need to set the visible property of the grid to "False".

```
Tasks .visible = False
```

Then you need to copy the calendar class from our example, and paste it in the bottom of the TaskCalendar\_Events.asp page.

This explains how you go about implementing the calendar class into any page that you want to display a dynamic calendar.

Let's revert back to the example code now and see how we get the calendar to execute and build our monthly view.

**Instantiating The Calendar Class**

Have a look at Figure #3 and you'll notice a few of the properties you need to set to control how the calendar displays it's data.

Remember that even though the CSS class indicates, "LightWalk" was used, we've created our own calendar.css file 'based' on a copy of the "LightWalk" style.css file.

But before you can go any further, you need to add a label to the page and add

some code to the BeforeShow() event of the label.

The labelname\_BeforeShow() event is where we instantiate the class and it's this label that 'becomes' the calendar.

You'll notice that we have created some additional styles for the calendar and in order for the calendar to dynamically display an event using a particular style, we've added another field to the event\_categories Table called "css\_class".

The benefit of doing this is to cause all events of a given category to display similarly. However, going back to our "Task Manager" example, you might – in that instance – choose to add such a field to the Priorites table thus causing all "Tasks" to be displayed in colors according to the priority of the task.

**Retrieving The Data**

We've created a generic function to derive a recordset, called;

```
Function SelectData(vSQL, vReturnDataRS)
```

Ensure that you copy this over to any calendar pages you make as well as the calendar class.

Also, remember to change the name of the data class being instantiated within this routine accordingly.

NOTE: You may need to change the name of the data class this line in this function;

```
From; Set cn=New clsDBIntranet
```

Example: If you are going to display "Tasks", you'd add a field called "css\_style" to the Priorities Table.

Create your own calendar page.

Add a grid, set it's datasource and make the grid not visible.

Add a label to the form and copy and paste the code from labelname\_beforeshow() in our example into the BeforeShow() event of your label.

Modify the SQL according to your needs. Pay special attention to the field names and modify them appropriately.

Modify the MyCalendar.OnDayClick property and the MyCalendar.DayViewPage property.

Modify the database class name in function SelectData().

Look in the .ASP code page (tab) to derive this class name. You should see this class instantiated somewhere around line 45 .

```
(
'Initialize Objects @1-CBD1C618
Set DBIntranet = New clsDBIntranet
)
```

Run the code. When you have the SQL right you can begin to modify the Styles.CSS styles according to your specifications.

Create Event Calendars, Task Calendars, Hotel Reservation Systems. etc.. Have Fun!

# Building A Dynamic Calendar Using CCS & PHP

## Dynamic Calendar Creation With CCS...

Language: .PHP  
Database: Any

I would like to start off by giving my thanks to DataObjx.Net magazine for the opportunity to write this article.

This is my first and hopefully not my last article for the CodeCharge software.

I felt it necessary to write about my experiences with CodeCharge software because I feel that it could benefit other programmers/software engineers out there who are learning about this great software.

While I have programmed for many years and with different programming languages, I don't know everything but I am comfortable with what I do know. I say this because as you go through this article if you have a better way of doing something feel free to drop me a email: mambobrown@mambobrown.com. I am always open to new ways of doing things.

### Let's get started.

I came up with this idea of creating this calendar for a few reasons;

- (1) I wanted to make a calendar that could be incorporated into CodeCharge studio without making any changes to the core program files
- (2) I wanted it to be versatile in that it would work with any table/query as long as it had a date field and finally
- (3) I love challenges (not to mention that a client requested a calendar built into the software).

One of my strong beliefs in programming is never to recreate the wheel.

I have quite a few PHP programming books not to mention that I check out tutorials at Zend.com and Hotscripts.com.

For this calendar I borrowed from a tutorial at Zend.com, which was written by Patrick Delin (Building

Dynamic Calendars using PHP).

I took his calendar and integrated it into CodeCharge.

The first challenge I ran into with designing the calendar was how was I going to get CodeCharge studio to create the calendar before the page was rendered to the client.

I tried putting the script in the Before Page Event but it did not work.

I even tried putting it in a Form without Data Properties but that did not work either.

After a couple of other tries I went to the CCS tutorial (<http://ccs.ath.cx>) website but it called for changes to the core files which I did not want to do.

I then scoured the CodeCharge forums which more or less pointed me back to the CCS tutorial website (CCS is a good place to get many tutorials on how to do things in CodeCharge studio).

At this point I had been programming for about 4 to 5 hours and I decided to take a break to clear my mind.

Clearing your mind when you have been focused on something for a long period of time is not a bad idea from time to time.

For me I just stopped programming for the day. That evening, while spending time with my family, it hit me like a ton of bricks.

I was so excited that I was ready to start programming at home but I decided to wait until I got to work that next day.

The next day, I put my idea into action.

I had remembered an old trick I used in MSAccess that helped me when I need to relate an object to data for no other reason than being able to view and use the object (hopefully this will make sense to you).

I created a dummy table (named dummy) with one field and record, which would never be updated.

It would be used to tie a data source to an object, which in this case was a Grid Form. This would allow me to execute the script before the Grid Form was made visible and assigned the generated html code to a Label Field in the Grid Form.

Did you get all of that ☺!

Before we start incorporating code let's state some obvious things that should already exist.

You should already have a table/query that has a date field included in it.

This will be used in generating the information that is related to a specific date.

You should already know what information will be displayed on the calendar.

In this case a Project Code (i.e. EB2314, LNGHRN) and Project Status (i.e. Deadline, Meeting) will be displayed.

Also, I decided to have the detail information of the project to pop up in a separate window.

The javascript (that creates the pop up screen) is included in the script.

Finally, the table/query should have separate screens where it can be updated with detail information.

### Let's get to the good stuff.

Like coding...

First, I created a page named maincalendar with a header and footer.

Then I added a grid form to the page from the Forms Tab in the Toolbox.

I configured the Grid Form to look like **figure 1**.

Next, I set the Grid Form Data Properties (*see figure 2*).

Moving on to the Label field in the form, I clicked on the Label tag once and then selected the Events Tab and added Custom Code to the Before Show Event (*see figure 3*).

I tried to comment the code as best I could (*see figure 4*).

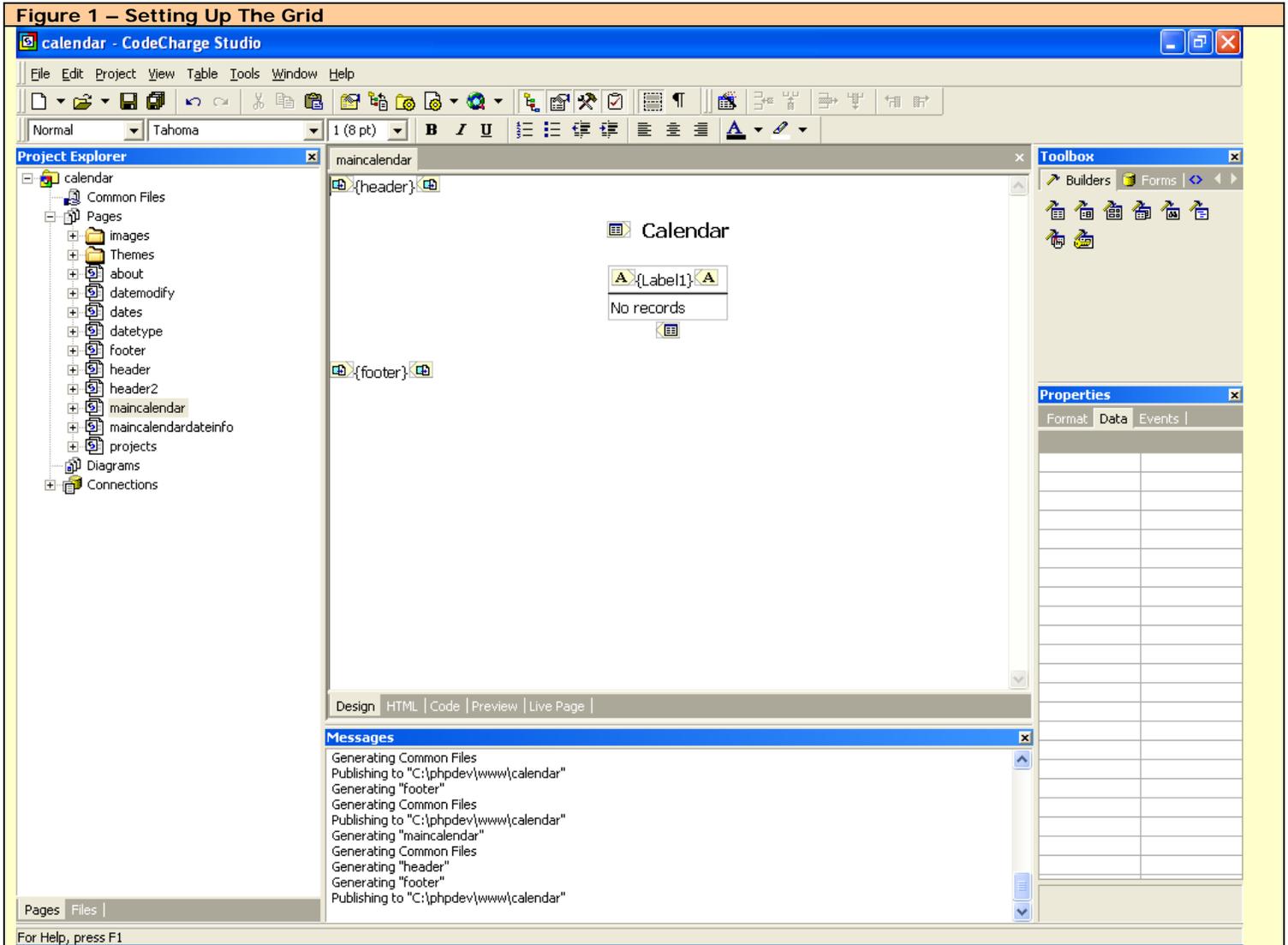
I truly hate commenting but it is necessary. Also, as you read the

# Building A Dynamic Calendar Using CCS & PHP

Language: .PHP  
Database: Any

Dynamic Calendar Creation With CCS...

Figure 1 – Setting Up The Grid



comments you will notice that there is a way to change the style of the calendar.

I could have done this better but because of time I left it as it was.

Finally, don't forget to create the pop up page for the project detail.

The javascript is generated by the calendar script in the maincalendar page.

In this example I created a page called maincalendardateinfo.

The page is a basic form with the project detail fields listed.

The page reads the parameters that are passed to it from the maincalendar page

## CCS MarketPlace Is Open For Business

SELL YOUR CCS PRODUCTS ON LINE

Sell Your Code On-Line

If you have been building or have already built a product using CodeCharge Studio that you think other CCS Developers would be interested in, place your product on the CCS MarketPlace.

**Contact Us Today!**

Send an email to [administrator@dataobjx.net](mailto:administrator@dataobjx.net) today and let us know what your application is all about. We'll place it on the CCS Marketplace and advertise it in CCS Developer Magazine.

**The CCS Marketplace**

Where you go to buy CCS Solutions!

# Building A Dynamic Calendar Using CCS & PHP

Language: .PHP  
Database: Any

## Dynamic Calendar Creation With CCS...

(see figure 5).

**Guess what? Your done!!!**

Now all you have to do is publish the project and give it a try.

If you would like to see it in action here is a link to give it a try <http://www.mambobrown.com>.

This calendar script was my first take.

I do plan to greatly improve on its' abilities but for now hopefully it can be of some use to you.

I also hope that I have given you some ideas on how you can use this method (dummy table assigned to a Grid Form) for integrating other scripts as well.

If you find the script useful please drop me an email (mambobrown@mambobrown.com). Thanks and enjoy.

### About Edwin Brown

Edwin Brown is a DBA/software engineer and web designer. (CCS bulletin board name - mambaBROWN)

During the day he works for the Federal Government and during the evenings runs his own web company in the capacity of CEO/President.

### WRITE FOR CCS DEVELOPER MAGAZINE

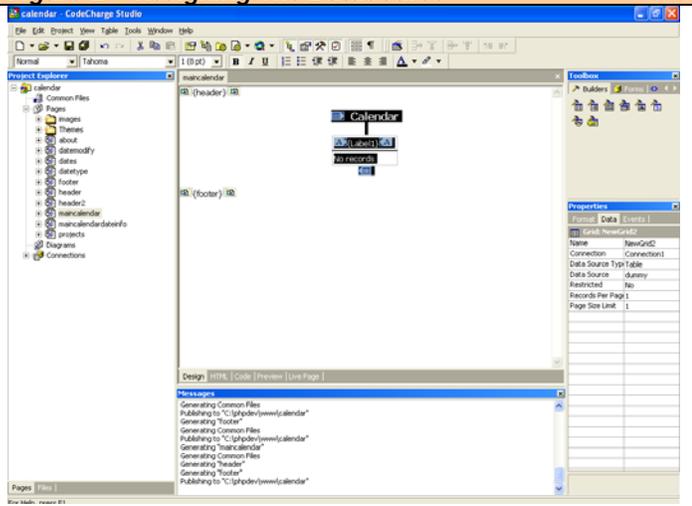
You can write articles for CCS Developer Magazine or just share your tips and tricks.

If English is not your first language or you don't think that you can write prose; don't let that stop you!

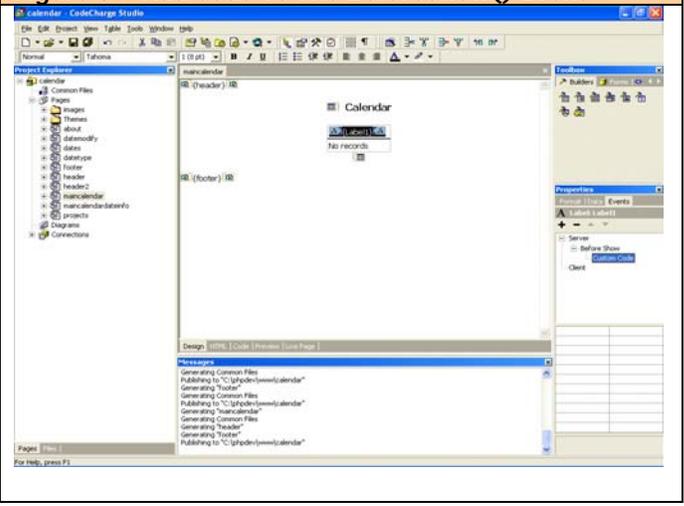
If you can supply the code and an explanation of what you're doing – we'll format it and publish it.

Send us an email about your article today!

**Figure 2 – Assigning The DataSource**



**Figure 3 – Add Code To The BeforeShow() Event**



**Figure 4 – The Calendar Code**

```
global $NewGrid2;

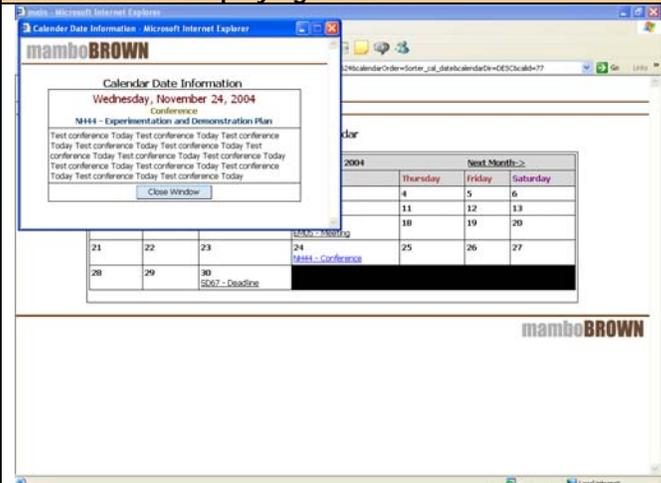
$db = new clsDBConnection1();
$db2 = new clsDBConnection1();

$monthtop=" <center> ";

//pull variables from url
$mnth = CCGetParam("m",0);
$yr = CCGetParam("y",0);

//determine if month and year is valid
if ((($mnt>0 and $mnt<13) and ($yr>1900))
{
    $date = getdate(mktime(0,0,0,$mnt,1,$yr));
}
else { $date = getdate(); }
```

**PHP Calendar Displaying An Event**





# Building A Dynamic Calendar Using CCS & PHP

Language: .PHP  
Database: ANY

## Dynamic Calendar Creation With CCS

Figure 4 – The Calendar Code, continued...

```
// begin placement of days according to their beginning weekday
$day = 1;
$wday = $first_week_day;
$firstweek = true;

while ( $day <= $lastday)
{
    if ($firstweek)
    {
        $monthtop=$monthtop."<TR valign=top>";
        for ($i=1; $i<=$first_week_day; $i++)
        {
            $monthtop=$monthtop."<TD> </td>";
        }
        $firstweek = false;
    }
    if ($wday==0) { $monthtop=$monthtop."<tr valign=top>"; }

    if ( intval($month_num) < 10) { $new_month_num = "0$month_num"; }
    elseif (intval($month_num) >= 10) { $new_month_num = $month_num; }
    if ( intval($day) < 10) { $new_day = "0$day"; }
    elseif (intval($day) >= 10) { $new_day = $day; }
    $link_date = "$year-$new_month_num-$new_day";

    $htmlcode = "";

    //find project calendar items for this day. If there is an item for the day then make
    //the item linkable to the popup detail page - maincalendardateinfo.php
    $db->query("SELECT calid, pjid, cal_datetype FROM calendar WHERE YEAR(cal_date)=". $year. "
and MONTH(cal_date)=". $month_num. " and DAYOFMONTH(cal_date)=". $day. " and markdelete_cal=1");
    $Result = $db->next_record();
    if ($Result)
    {
        $alinkref = $db->f("calid");
        $apjid = $db->f("pjid");
        $acaltypeid = $db->f("cal_datetype");
        $aabbrv = CCDLookUp("pj_name_abbrv", "projects", "pjid=". $apjid, $db2);
        $acaltype = CCDLookUp("dt_datetype", "datetype", "dtid=". $acaltypeid, $db2);
        $htmlcode = $htmlcode. "<a href=\\"javascript: void(window.open(
'maincalendardateinfo.php?calid=". $alinkref. "\", 'DateInfo', 'toolbar=no, location=no, status=no,
menubar=no, scrollbars=yes, resizable=no, width=503, height=300'))\">". $aabbrv. " - ". $acaltype. "</a><br>";

        while ( $db->next_record() )
        {
            $alinkref = $db->f("calid");
            $apjid = $db->f("pjid");
            $acaltypeid = $db->f("cal_datetype");
            $aabbrv = CCDLookUp("pj_name_abbrv", "projects", "pjid=". $apjid, $db2);
            $acaltype = CCDLookUp("dt_datetype", "datetype", "dtid=". $acaltypeid, $db2);
            $htmlcode = $htmlcode. "<a
href=\\"javascript: void(window.open('maincalendardateinfo.php?calid=". $alinkref. "\", 'DateInfo', 'toolbar=no,
location=no, status=no, menubar=no, scrollbars=yes, resizable=no, width=503, height=300'))\">". $aabbrv. " -
". $acaltype. "</a><br>";
        }
    }
    $monthtop=$monthtop."<td class=\\"Graphite2DataTD\"><b> $day </b><br>". $htmlcode. "</td>";
    if ($wday==6) { $monthtop=$monthtop."<tr>\n"; }
    $wday++;
    $wday = $wday % 7;
    $day++;
}
$monthtop=$monthtop."</table></center>";

//Set Label1 field to generated calendar html code variable $monthtop
$NewGrid2->Label1->SetValue($monthtop);
// -----
//End Custom Code
```

# RSS 2.0 Feed Generator Functions

Language: .PHP  
Database: Any

## Creating An RSS.XML Feed For Your Site

Article Contributed by: Femi Hasani

### RSS 2.0 FEED Generator Function

We will not elaborate in this article about RSS feeds as there is a ocean of resources about RSS on the Internet.

#### “Do I need a RSS feed?”

Well, these days there are many FREE RSS FEED Readers on the internet.

Since many people do not have much time to visit each of their favorite sites to check for updates they simply insert their RSS feeds of interest and let the RSS reader check for updates.

The other benefit is the sites can use your RSS feeds to parse and view latest updates of your articles on their websites.

We have created a function that will generate a rss.xml file after each update of your records.

This creates a “static” rss.xml file instead of a “dynamic” rss.xml file.

The difference is not a “big” one but if we were to use a “dynamic” (real-time) creation of RSS feeds there would be un-necessary database queries for each request whether you have updated your records or not.

This would overload the server without need *especially* if 1000’s of users would use your feed at the same time.

The “static” file will updated only when you update your records.

#### Validating Your RSS.XML File

To check if your generated FEED is valid you can use this FREE validator located at;

<http://feedvalidator.org/check.cgi?url=http://www.vision.to/new cms/Com mon/rss.xml>

You have to take care about using special characters in your description text this may make your feed invalid.

#### The Code For The Generator

The complete function is included in this zip file with detailed documentation with in the code.

The picture in Figure 1 will explain how to call this function when your record is being updated

Figure 1 – Function AfterUpdate() Event

```

201 function ddd_codel_AfterUpdate()
202 {
203     $ddd_codel_AfterUpdate = true;
204     //End ddd_codel_AfterUpdate
205
206     //Custom Code @229-A5D0817F
207     // -----
208     global $ddd_codel;
209     // Write your own code here.
210     if($ddd_codel->update_rss->GetValue()==1)
211     {
212         $report=create_codelib_rss_xml();
213     }
214
215
216     // -----
217     //End Custom Code
218
219     //Close ddd_codel_AfterUpdate @61-62470BFC
220     return $ddd_codel_AfterUpdate;
221 }
222 //End Close ddd_codel_AfterUpdate
223
224
225
226
227
    
```

You will need to call the function in an AFTER UPDATE event. In this sample \$ddd\_codel1->update\_rss->GetValue() is an check box so if checked it will update RSS file else not.  
(Note: The name of the function is changed for the particular record)

Real Example:

[http://www.vision.to/new cms/CodeLib\\_MOD/index.php](http://www.vision.to/new cms/CodeLib_MOD/index.php)

FEED

[http://www.vision.to/new cms/CodeLib\\_MOD/rss.xml](http://www.vision.to/new cms/CodeLib_MOD/rss.xml)

VALIDATOR

[http://feedvalidator.org/check.cgi?url=http://www.vision.to/new cms/CodeLib\\_MOD/rss.xml](http://feedvalidator.org/check.cgi?url=http://www.vision.to/new cms/CodeLib_MOD/rss.xml)

Hope this will help you much to enhance your web applications.  
(I’m missing a RSS FEED on CodeCharge Studio FORUM ☺)

Figure 2

```

<?php
/*
<!--
*****
*       Website: http://www.vision.to           *
*       Copyright © by WWW.VISION.TO           *
*       All Rights Reserved                     *
*       Author: Femi Hasani                     *
*       Contact: info@vision.to                 *
*       This application is protected by international *
*       copyright laws.                         *
*****
Version 1.1 Special Release for CCS Developer Magazine
FOR USE FREE OF CHARGE YOU MUST KEEP (VISIBLE)
    
```

# RSS 2.0 Feed Generator Functions

## Creating An RSS.XML Feed For Your Site

Language: .PHP  
Database: Any

Figure 2 – Continued...

```
<generator>www.vision.to v 1.0 Vision.To CMS Enterprise</generator>
In your generated XML file.
Copyright statement above must remain intact.
LATEST UPDATES RSS FEED
This function will create/generate a rss.xml RSS2.0 compatible FEED from your DB table.
-->
*/
// IMPORTANT
//for use with CodeCharge Studio include this file at the end of Common.php
//will check if you have defined before some constants.
if(!defined("LANG_CODE"))
{
define("LANG_CODE","en-us");
}

if(!defined("XML_ENCODING"))
{
define("XML_ENCODING","UTF-8");
}

if(!defined("COMPANYNAME"))
{
define("COMPANYNAME","Your company-website name/title");
}

if(!defined("COMPANY_START_YEAR"))
{
define("COMPANY_START_YEAR","1998");
}

if(!defined("CUSTOM_TIME"))
{
define("CUSTOM_TIME",time());
}

if(!defined("CONTACT_E_MAIL"))
{
define("CONTACT_E_MAIL","webmaster@yoursite.com");
}

if(!defined("URL_PATH"))
{
define("URL_PATH","http://www.vision.to");//Replace with your's
}

//Will define if not defined the location of your RSS LOGO visible to RSS Readers.
if(!defined("XML_LOGO"))
{
define("XML_LOGO","http://".$_SERVER['SERVER_NAME']."".dirname(str_replace("\\", "/",
$_SERVER['SCRIPT_NAME']))."/rsslogo.png");
}

//call this function as create_rss_xml_feed();

function create_rss_xml_feed()
{
$RSS_XML = "";
$RSS_XML .= "<?xml version=\\"1.0\\" encoding=\\"".XML_ENCODING."\"?>\n";
$RSS_XML .= "<rss version=\\"2.0\\">\n";
$RSS_XML .= "<channel>\n";
$RSS_XML .= "<title>".htmlspecialchars(COMPANYNAME)."- CodeLibrary RSS Feed</title>\n";
$RSS_XML .= "<link>".URL_PATH."</link>\n";

$RSS_XML .= "<description>".htmlspecialchars(COMPANYNAME)."- Homepage, 20 Latest CodeLibrary Updates RSS 2.0
Feed</description>\n";
```

### Services Provided By Vision.To Design

#### Services

- Web Design
- Custom PHP programming
- Database Design (MySQL)
- CodeCharge Studio & CodeCharge Based Projects
- Graphics Design
- Logo Design
- Dynamic Flash content (Swishmax + PHP + MySQL)
- Multimedia Web production, etc.
- Modular Design
- Special portals for Your Internet/Intranet
- e-commerce solutions

\* Custom web applications for Your needs exactly as You want it.

\* All of our web applications include web administration.

-----  
Vision.To Design  
Femi Hasani  
[www.vision.to](http://www.vision.to)  
[feha@vision.to](mailto:feha@vision.to)  
+46 736 009 555  
Sweden / Suedi  
[www.coder24.com](http://www.coder24.com)  
[www.wowdev.com](http://www.wowdev.com)

# RSS 2.0 Feed Generator Functions

Language: .PHP  
Data Base: Any

## Creating An RSS.XML Feed For Your Site

### Figure 2 Continued...

```
// IMPORTANT
//You need to modify to suit your connection name
$DBddd_rss = new clsDBddd();
//You need to modify to suit your table name and columns
$SQL = "SELECT * FROM ddd_code WHERE code_approved = 1 AND private <> 1 AND user_level <= 1 ORDER BY code_updated
DESC LIMIT 20";

//$SQL = "SELECT * FROM ddd_pages WHERE page_show =1 AND no_menu=0 AND user_level=0 ORDER BY page_edited DESC
LIMIT 10";

$DBddd_rss->query($SQL);
while ($DBddd_rss->next_record()) {
$RSS_XML .= "<item>\n";
$RSS_XML .= "<title>\n";

//You need to modify to suit your table column name
$RSS_XML .= htmlspecialchars($DBddd_rss->f("code_name"));
$RSS_XML .= "</title>\n";
$RSS_XML .= "<description>\n";

//$RSS_XML .= "<![CDATA[ \n";
//You need to modify to suit your table column name
$RSS_XML .= htmlspecialchars($DBddd_rss->f("code_desc"));
//$RSS_XML .= $DBddd_rss->f("code_desc");
//$RSS_XML .= "\n]]>\n";
$RSS_XML .= "</description>\n";

//You need to modify to suit your table column and file index.php?code_id name
$RSS_XML .= "<link>http://".$_SERVER['SERVER_NAME']."".dirname(str_replace("\\", "/",
$_SERVER['SCRIPT_NAME']))."/index.php?code_id=".$_DBddd_rss->f("code_id")."</link>\n";

//You need to modify to suit your table column name
$RSS_XML .= "<pubDate>".date("r", $DBddd_rss->f("code_updated"))."</pubDate>\n";
$RSS_XML .= "</item>\n\n";
}
$DBddd_rss->close();
$RSS_XML .= "</channel>\n";
$RSS_XML .= "</rss>\n";

//echo $RSS_XML; //DEBUG
//After XML string is created we need to create and put it in to the rss.xml file
//IMPORTANT: the directory has to have premissions CHMOD 777
//This file will be created under the same directory the function has been called form particular
// page containing the record_form within that directory
$report="";
$filename = "rss.xml"; //define desired file name for your RSS FEED

if (!file_exists($filename)) {
touch($filename); // Create blank file rss.xml
}
if (is_writable($filename)) {
if (!$handle = fopen($filename, 'w+')) {
$report="Cannot open file ($filename)";
}
if (!fwrite($handle, $RSS_XML)) {
$report="Cannot write to file ($filename)";
}
}
$report="Success, ($filename) has been created, updated ";
fclose($handle);
} else {
$report="The file $filename is not writable";
}
//The file should have been created by now ...
return $report;
}
?>
```

# Automating Excel Report Generation

Language: .ASP  
Database: Any

## Creating Excel Files Based On A SQL Query...

It's tough out there. People want data for various reasons and we are in the business of capturing data. That's why we use CodeCharge Studio to generate our database driven applications with great success.

But our web applications don't always perform all of the actions that the owner of the data might wish for.

Sometimes, the data was captured for one reason but is valuable for other reasons.

The code that we're going to create in this article will enable you to provide "Canned Reports" as well as dynamic reports – output to an Excel Spreadsheet.

### Let's Have A Look

Open the project file and in the project settings, configure the project data source to point to a database on your development server. We'll call this your "portal".

### Set the Publish Directory To "Reports"

For the publish path, set the project to point to "c:\inetpub\wwwroot\Portal\Reports"

You may also need to set file Read/Write permissions to IUSR\_... or IWAM\_...

Once you've done this, re-generate the application.

### The ForceDownload Page

Contained within the .zip file for this project is another page called "ForcedDwnload.asp".

The "ForcedDwnload" Page has been changed to an 'include' file, even though it's not "included" anywhere and acts as a stand alone page.

The page never displays anything however. Rather, it 'forces' a download dialog to the user.

Note: If you change this from an "Includable File" to a normal file, you will get a "Name Redefined" error. That's because the ForceDownload\_Events.asp page already contains a reference to the ADO. So leave it as an include.

### The CreateExcelFile Page

Now, open the CreateExcelFile pages.

The CreateExcelFile Page has been changed to an 'include' file.

The code within the CreateExcelFile\_events.asp page does just that, it creates an Excel file.

### The Default Page

The default page is where all of the action occurs as it captures and passes to the CreateExcelFile routines, those parameters necessary to cause the Excel file to be created.

The default page also has a 'Record' on it though you'll notice that no datasource has been set.

We don't need to set a datasource for this project however. Instead, the 'record' provides us with a DataSource and that's the only reason it's there.

### Two (2) Modes Of Operation

The default page has 2 modes.

#### Automated Report Mode

One mode allows for automated report generation and delivery. This is the 'automated report' method.

Automated mode requires that a "?report=REPORTNAME" is present in the querystring.

#### DBA Mode

The other mode allows a person with sufficient rights to access the SQL Text Box interface. That's for the DBA or System Administrator.

DBA Mode occurs when "?report=REPORTNAME" is NOT present in the querystring.

You may want to make this into two (2) separate pages for security reasons. For example, copy and save the "Default" page as "DefaultDBA" and then set the security to the DBA/Admin level.

You would then remove the Title and Text Box from the original "Default" page. (But leave the 'record' on the page).

We don't want to muddy up the water though by talking about two (2) different pages, so we'll leave the breaking up (into two pages) to you.

### Concept

Before we go any further, you need to understand the concept as it will make it easier for you to understand where this is going and moreover – where you can take it.

DBA Mode is just that. It allows a person familiar with the data schema to execute SQL commands against the database. That includes views and stored procedures.

The Automated Report Mode is the one that will be used the most and that provides you with an opportunity to create a really great reporting tool.

### How To Change Modes

If you haven't already guessed, changing modes is the difference between having "?report=REPORTNAME" in the querystring or not.

Take a moment to execute the default page in each mode.

```
.../default.asp?report=events
and
.../default.asp
```

Note: you may need to modify the SQL in the Page\_BeforeShow() event to correlate to your data schema.

### Canned Reports

If you open Default and look in the Page\_BeforeShow() event, you'll notice that we've hard coded a few SQL queries into a select case statement.

If the querystring contains a "?report=REPORTNAME" in it, the code will cause a variable called (what else?) ReportName.

```
ReportName =
UCase(Request.QueryString("Report
Name"))
```

This is then compared against the Select Case statement and the SQL is derived accordingly.

A Title for the Spreadsheet is also assigned at this stage.

Then the code performs a call to the function CreateSpreadsheet(gTitleForSpread sheet, gSQL)

# Automating Excel Report Generation

## Creating Excel Files Based On A SQL Query, continued...

The CreateSpreadsheet function is located in the CreateExcelFile include.

It's purpose is to take the recordset resulting from the SQL Query and transform it into an Excel Spreadsheet and save it to the "Reports" directory.

Finally, it redirects to the ForceDownload page passing the path and file name and causing a download dialog to appear for the user.

### Extending The Application

What benefit to you derive from this aside from automating the generation and downloading of an Excel Spreadsheet file?

Well, depending on how many users are within your organization or the number of reports you are manually creating... it could be quite a lot.

The Select Case statement should have immediately clued you in. This is a perfect scenario where-in you could derive the SQL from a database table, based on the name of the report being passed.

Your table structure may look like this;

```
Table ReportsSQL
Id, int @@IDENTITY
Report_Name varchar(40)
Report_Title varchar(100)
Report_SQL (255)
```

Now, you'd simply perform a CCDLookup on the ReportsSQL table where the Report\_Name = Request.QueryString("Report") and you'd have the Title and SQL you need to pass to the function.

### Applying The Technology

Next time an employee needs a special subset of data, you'll be able to set up the query and provide your user with a link to get the report again whenever they want.

```
Function CreateSpreadsheet
Function CreateSpreadsheet(vSpreadsheetTitle, vSQL)
On Error Resume Next
Dim fsoExcel
Dim rs
Dim SQL
Dim fso
Dim i
Dim longBinaryText
Dim dteDate

g_filename = vSpreadsheetTitle

g_filename = g_filename & "-" & month(date()) & day(date()) & year(date()) & ".xls"

set rs = createobject("ADODB.Recordset")

If Not SelectData(vSQL, rs) Then
    If rs.eof then
        Response.Write "Unable to derive recordset to build spreadsheet"
        Response.end
    End If
End If

set fso = createobject("scripting.filesystemobject")
' create the text (xls) file to the server adding the -mmddyyyy after the g_title value
Set fsoExcel = fso.CreateTextFile(server.mappath("'" & g_filename), true)

' write all of the user input to the text (xls) document
fsoExcel.WriteLine "<html xmlns:x=""urn:schemas-microsoft-com:office:excel"">"
fsoExcel.WriteLine "<head>"
fsoExcel.WriteLine "<!--[if gte mso 9]><xml>"
fsoExcel.WriteLine "<x:ExcelWorkbook>"
fsoExcel.WriteLine "<x:ExcelWorksheets>"
fsoExcel.WriteLine "<x:ExcelWorksheet>"
fsoExcel.WriteLine "<x:Name>Sheet1</x:Name>"
fsoExcel.WriteLine "<x:WorksheetOptions>"
fsoExcel.WriteLine "<x:Print>"
fsoExcel.WriteLine "<x:ValidPrinterInfo/>"
fsoExcel.WriteLine "</x:Print>"
fsoExcel.WriteLine "</x:WorksheetOptions>"
fsoExcel.WriteLine "</x:ExcelWorksheet>"
fsoExcel.WriteLine "</x:ExcelWorksheets>"
fsoExcel.WriteLine "</x:ExcelWorkbook>"
fsoExcel.WriteLine "</xml>"
fsoExcel.WriteLine "<![endif]-->"
fsoExcel.WriteLine "</head>"
fsoExcel.WriteLine "<body>"
fsoExcel.WriteLine "<table>"

Do While Not rs.EOF
    fsoExcel.WriteLine "<tr>"
    For i = 0 To rs.fields.count-1
        fsoExcel.WriteLine "<td>"
        fsoExcel.WriteLine rs.Fields(i).name
        fsoExcel.WriteLine "</td>"
    Next
    fsoExcel.WriteLine "</tr>"
Exit Do
Loop
rs.MoveFirst

'CONTINUED ON NEXT PAGE
```

# Automating Excel Report Generation

## Creating Excel Files Based On A SQL Query...

### Function CreateSpreadsheet, Continued...

```

Do While Not rs.EOF
    fsoExcel.WriteLine "<tr>"
    For i = 0 To rs.fields.count-1
        fsoExcel.WriteLine "<td>"
        'BLOB/IMAGE/TEXT FIELDS NOT ALLOWED
        'You may choose to perform additional editing on various fields as well,
        'For example.
        'Field Type 203: Text Field - Requires NeatTrim()
        'Field Type 135: Date Fields - Requires Formatting

        If rs.Fields(i).type <> 3 AND rs.Fields(i).type <> 135 AND rs.Fields(i).type <> 203 Then
            fsoExcel.WriteLine rs.Fields(i) & ""

        ElseIf rs.Fields(i).type = 3 Then
            fsoExcel.WriteLine CStr(rs.Fields(i)) & ""

        ElseIf rs.Fields(i).type = 135 Then
            If IsDate(rs.Fields(i)) Then
                dteDate = FormatDateTime(rs.Fields(i)) & ""
            Else
                dteDate = ""
            End If
            fsoExcel.WriteLine dteDate

        ElseIf rs.Fields(i).type = 203 Then
            longBinaryText = neatTrim( rs.Fields(i) & "", 65000 )
            fsoExcel.WriteLine longBinaryText & "Long Binary..."
        End If
        fsoExcel.WriteLine "</td>"

        If Err.Number <> 0 Then
            Response.write "Absolute Position: " & rs.AbsolutePosition & "<BR>"
            Response.write "Failed On Field Name: " & rs.Fields(i).name & "<BR>"
            Response.write "Failed On Field Type: " & rs.Fields(i).type & "<BR>"
            Response.write "Failed On Field Value: -->" & rs.Fields(i) & "<---<BR>"
            'Response.end
        End If
    Next
    fsoExcel.WriteLine "</tr>"
    rs.MoveNext
Loop

fsoExcel.WriteLine "</table>"
fsoExcel.WriteLine "</body>"
fsoExcel.WriteLine "</html>"
' close the document
fsoExcel.close

'stream the file to the user
Response.redirect "forcedownload.asp?file=" & Server.MapPath(".") & "\" & "&name_of_file=" & g_filename

End Function

```

### The CCS MarketPlace Wants Your Application

#### Open For Business

The CCS MarketPlace is there for you to sell your CCS Projects.

Brush off that application and contact [administrator@dataobjx.net](mailto:administrator@dataobjx.net) to get your application on-line today! Why wait. Make money from your efforts!