CCS Developer

SEPTEMBER 2004

VOLUME NO. 1 ISSUE NO. 2

DataObjx L.L.C.

Website: www.dataobjx.net

DataObjx L.L.C.
Meriden, CT USA

Office: 860/919-2536

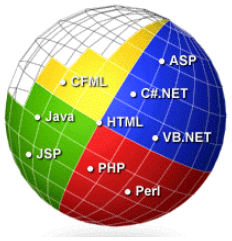
Email: administrator@dataobjx.net

Website: www.dataobjx.net

DataObjx L.L.C.

Do you have an article to contribute?

Contact DataObjx to find out how.



Need More Answers?
Click here to access the Code Charge Studio Forums

In This Issue

Editors Comments

Viewpoint...

First, we'd like to thank all of you who downloaded our first issue and those of you who contributed their comments and articles for publication.

We've been working very hard on both the DataObjx website and the magazine and as the DataObjx website becomes complete – we'll be able to focus more on our core business – providing you with a better CCS Developer Magazine!

We have to tell you – it's been tough. Building the Web site, writing the applications for the articles and generally 'gearing up' after the launch has been a difficult task – every step of the way.

We launched the CCS Jobs Board and we need each of you to update your language and database skills in your Member profile so that we can make the Jobs Board more effective.

We have also been struggling with the PayPal IPN system so that you can get the magazine immediately after purchase. As you may know, PayPal has been taken over by eBay and there have been a number of changes and increased complexity in some cases and this has cause us to burn more hours than we'd have liked. But we wanted to try to make sure that you could download the magazine after your purchase.

We also added a CCS Web Sites Resource list – so you can let others know that you're out there. So support the CCS community by placing your Web Site information on the CCS Web Sites listing – next time you login.

We're also trying to provide you with applications that far out-weigh the price of the magazine. For instance, by our 3rd issue you'll have a Document management system that literally cost you about $10 USD. Not bad considering the tens of hours it took to write the application. So, we're trying our best to provide you with value for your money and simply ask that you bear with us as we improve with each issue.

In our second issue, we continue to build on our document management system. And in our 3rd issue we'll added more capabilities (such as "Check-In/Check-Out" capabilities) as well as other features and began to refine our wizard interfaces and finishing touches.

We wanted to provide you with a look at the ASP Report Wizard by http://www.aspwebsolution.com/ but this article has been moved to the October issue.

We also began a new series called 'Expanding The CodeCharge Studio Portal' where we'll begin to show you ways you can improve your web site or intranet.

As a bonus, we're going to provide a sneak peek at DemoCharge, the latest product by YesSoftware, Inc.

We've also listened to your comments and we hope that you find this issue more streamlined and easier to follow.

We've also had a number of developers using languages other than asp respond and we're encouraged by the promise of articles that may be of more direct interest to you. In fact, in this issue, we provide you with an article that utilizes PHP and Oracle.

Moreover, you'll also notice a change in how we deliver the SQL for the applications. Since so many of you are using SQL Server, Oracle and MySQL databases for your applications – we will be trying to provide pure SQL rather than MS Access examples. Where possible, we'll still provide the MS Access Databases however -even though most of you end up having to transfer the data schema to your higher-end database anyway. But by providing the data schema, stored procedures and views along with the magazine we hope it makes it easier for you to scale the code to your preferred database.

We've also been adding a number of applications and documents to the 'Resources/Tools" area – including a small application called SQLFormatter that converts your raw SQL into SQL Strings that are ready to be plugged directly into your code.

So, Update your member profile – languages and database skills, download the documents and tools located in the Resources area and send us your articles and comments.

We hope you enjoy issue #2.

Best Regards,

Martin Hamilton
www.dataobjx.net

Language: .ASP
Database: SQL Server

Expanding The CodeCharge Studio Portal – Part 1

Leting Your Users Know If They've Read The Article Or Not.

Expanding The Articles Section

In our last issue, we demonstrated a way to use the label component to create dynamic row colors based on database values.

In this article, we're going to apply this method to a real world application.

If you build intranets or you have an Internet site requiring a login, this article gives you the ability to provide your users with a new feature.

Since the CodeCharge **Portal** application is widely used, we're going to use this application as the basis for our example.

YesSoftware made it easy for us to build intranets and portals when they supplied this application as an example.

We're going to focus on the Articles Table/Area in this issue.

We will add a number of tables so that we don't interfere with any existing code that may be driving your web site application.

As we continue this series, we're going to demonstrate new techniques. If you want to implement them into your existing application, we suggest that you build the new pages we discuss in these articles.

Then you will connect these pages when they are working as anticipated.

The CodeCharge Studio 'Portal' contains an "Articles" table.

We're going to provide you with the concept and code necessary to allow your users to spot new articles vs. articles they've already read.

Naturally, you will find other ways to use this concept in your applications.

Remember though - the technique demonstrated in this article works only with applications where the user is required to login to your application.

Therefore, let's begin by creating four (4) new tables.

Rather than use the "Articles" table – we're going to create a new table called "Portal_Articles".

We are also going to create a new categories table called "Portal_Article_Categories".

A table called Portal_Articles_Read.

And finally, a table called "Portal_Articles_Priorities"

The First Set Of Features

Authorized persons add and update articles on the Portal at various times.

When an authorized users logs in, there may be a number of new articles and a number of old articles.

We've probably relied on the order in which the articles are displayed by setting an Order By clause based on the date_added field, thus forcing newer articles to the top of the list – and old ones to the bottom.

In our new version, the sort order of the articles will be dictated by;

- The date_added field, and
- The priority field

In other words, if there is more than one record on any given date, the article with the highest priority appears at the top of the grid.

Now that we've addressed the order in which articles will appear in the grid, lets add some automation to the articles.

Allowing Articles To Expire

Over time we'd begin to see another problem that we need to address. In it's current state, the site requires constant "Content Maintenance" – E.g., the content becomes 'out of date' and manually needs to be deleted, etc.

For example, an article that announces the Office Party at the end of the month is no longer valid after the party is over.

If we don't want the article to appear after the party is over, someone with administrator rights needs to login and delete the article. The system therefore requires constant content maintenance. So what we're going to do is 'automate' the day that articles will be visible in the grid and the day that they 'automatically' disappear from the grid.

Knowing When An Article Has Been Read & Allowing The User To Archive The Article

We're also going to add the ability for the portal to record when a person reads an article and we're going to allow the person to 'Archive' any articles they like, removing them from primary view.

If you were building a 'Outlook' type of application the user could easily determine what mail they've read vs. new mail using the technique we'll demonstrate.

The technique of providing dynamic row colors explained in our first issue will come in hand later in this article.

Now that you understand what the new features are going to be and what the new tables are: let's begin.

Remember, we're creating new tables because we want to introduce new capabilities without affecting any existing code operating on our site. We'll be modifying our new tables in future issues.

Establish a connection to your database.

Those of you using SQL Server, Oracle, MySQL and so forth will want to open the appropriate consol needed to add and modify tables.

The Portal_Articles Table

Create the Portal_Articles table and add the following fields;

- Id As Integer, Primary Key
- Title As VarChar(255)
- Cat_ID As Integer
- Priority_ID As Integer
- Article_Teaser As VarChar(255)
- Article_Body As VarChar(255)
- Date_Begins As SmallDateTime
- Date_Ends As SmallDateTime

Expanding The CodeCharge Studio Portal – Part 1

Let Your Users Know If They've Read The Article Or Not, continued…

Language: .ASP
Database: SQL Server

- Publish_Article As Bit/Boolean
- Added_By_ID As Integer
- Date_Added As SmallDateTime

That's it for the Articles table for now.

The Portal_Article_Categories Table

Now we'll create the Portal_Article_Categories table. This is very similar to the event_categories table and for now we'll use a similar structure.

- Cat_ID As Integer, Primary Key
- Cat_name As Varchar(100)
- Added_By_ID As Integer
- Date_Added As SmallDateTime
- Publish As Bit/Boolean

The Portal_Article_Priorities Table

Now let's create the Portal_Article_Priorities table. This is going to hold various values to indicate the degree of importance an article has.

- ID As Integer, Primary Key
- Priority_name As Varchar(100)
- Added_By_ID As Integer
- Date_Added As SmallDateTime
- Sort_Order As Integer

The Portal_Articles_Read Table

Now lets create the Portal_Articles_Read table.

From your database consol, perform a Create Table command and call the table Portal_Articles_Read.

Add the following fields to the table;

- ID As Integer PRIMARY KEY
- User_ID As Integer
- Article_ID As integer
- Date_Accessed As SmallDateTime
- Archived As Bit

Summary Of Work

The Articles table now allows an administrator to 'tell' the Portal when an article can begin and when it should end and whether the article is visible (published).

The data schema now contains a priority id for the article to allow us to sort the articles grid in a more logical manner.

What's The Publish Button For?

We've included a "Publish" bit flag (Check box) that must be ticked or set to True in order for the article to be visible in the grid.

The reason this feature is handy is because it allows users to create an article over a period of hours, days or even weeks while it's visible only to the author.

When they place a check mark in the "Publish" tick box, the article only then becomes visible within the grid.

Modifying The Article Title

The articles screens are composed of a grid showing a listing of the articles and a read only page (Portal_Articles_View.ccp) that the end-user sees when they click on the title of the article.

The title has been created using a label control because we may make the title 'dynamic' in nature, in a future article.

The Data Source For The Grid

Open the project and select the portal_articles.ccp file. Open the data source for the grid.

We are using a stored procedure for the grid "sa_Portal_Articles_UnRead".

If you're using MS Access, you can create a Query and plug in the code you'll find near the end of this article.

This stored procedure uses SQL against a view (Query) to display only those article that are 'published' and that have not been 'Archived' by the

Figure 1 - Function portal_articles_article_title_BeforeShow()

```
Dim lngID
Dim sResult
Dim lngAuthorID
Dim strTitle
Dim strBody
Dim sStyle
Dim lResult

        lResult             = 0
        sResult             = ""
        sStyle              = "TitleLink"
        lngAuthorID         = portal_articles.datasource.recordset.fields("added_by_id")
        strTitle  = portal_articles.datasource.recordset.fields("article_title")
        lngID     = portal_articles.datasource.recordset.fields("article_id")
        If lngID > 0 Then
                lResult   = CCDLookup("id", "portal_articles_read", "user_id=" &
Session("PortalUserID") & " AND article_id = " & lngID & "", DBInternet)
                If lResult > 0 Then sStyle = "ReadTitleLink"
        End If
        sResult  = "<a href=""./portal_article_view.asp?article_id=" & lngID & """
class=""" & sStyle & """>" & strTitle & "</a>  "
        portal_articles.article_title.value = sResult
```

Expanding The CodeCharge Studio Portal – Part 1

Let Your Users Know If They've Read The Article Or Not, continued…

Language: .ASP
Database: SQL Server

user.

In addition, you will also notice that below the title is a label to display the value from the Article_Teaser field.

The purpose of this field is to provide the reader with a "Who, What, When & Where" synopsis of what the article is about.

This is a small 255 character field so brevity is necessary. This generally should not be a rich text field.

If you make this field a text/memo field and hook up a WYSIWYG editor, users may populate this area with a 10 page report not realizing that it will cause the grid to scroll on and on.

Therefore, make sure you write a little paragraph explaining that this field is for a synopsis from the author and is not the 'Body' of the article.

We've also created a page called "Portal_Articles_Archived.ccp" that contains a grid that displays the title of "Archived" articles. You will want to connect a link to your menu so that users can access this page.

This grid has been given a filter to cause the grid to ignore any articles that do not have the "Publish" flag set to true as well.

Modifying The Article_View Page

This is a read only page that displays the entire article, including the Intro_Teaser description.

If you look at the source for this page, you'll notice that we've added an additional function to update the Articles_Read table and this function is called in the Before_Show event for

Figure 2 - Function portal_articles_lblEdit_BeforeShow()

```
Dim lngID
Dim sResult
Dim lngAuthorID
Dim strTitle

lngAuthorID       = portal_articles.datasource.recordset.fields("added_by_id")
strTitle   = portal_articles.datasource.recordset.fields("article_title")
sResult              = ""

If Len(Session("PortalUserID")) > 0 Then
If CLng(Session("PortalUserID")) = lngAuthorID Then
lngID     = portal_articles.datasource.recordset.fields("article_id")
sResult   = "<a href=""./portal_articles_maint.asp?article_id=" & lngID &
""""><image border=""0"" src=""http://www.dataobjx.net/images/script.gif""
title=""Edit Your Document""></a>  "
End If
End If
portal_articles.lblEdit.value = sResult
```

Figure 3 - Portal_Articles_View - Function portal_articles_BeforeShow()

```
Dim SQL
Dim Connection
Dim ErrorMessage
Dim lID
Dim sResult
Dim ccSQL

SQL = ""
CcSQL = ""
lID = Request.Querystring("article_id")
If Len(lID) = 0 Then Response.Redirect "Portal_Articles.asp"
  CcSQL = "user_id=" & Session("PortalUserID") & " AND article_id = " & lID & ""
  '//First, let's see whether the user has already accessed the article.
  sResult = CCDLookup("id", "portal_articles_read", ccSQL, DBInternet)

  '// If they haven't, add a record to the portal_articles_read table.
  If Len(sResult) = 0 Then
    SQL = SQL & "INSERT INTO portal_articles_read (user_id,article_id,date_accessed) "
    SQL = SQL & "VALUES ("
    SQL = SQL & Session("PortalUserID") & "," & lID & ",'" & FormatDateTime(Now(), vbShortDate) & "'"
    SQL = SQL & ")"
  End If
Set Connection = New clsDBInternet
Connection.Open
Connection.Execute(SQL)
ErrorMessage = CCProcessError(Connection)
Connection.Close
Set Connection = Nothing
On Error Goto 0
```

Expanding The CodeCharge Studio Portal – Part 1

Let Your Users Know If They've Read The Article Or Not, continued…

Language: .ASP
Database: SQL Server

Figure 4 - Function portal_articles_lblStyle_BeforeShow()

```
Dim lID
Dim lResult
Dim sStyle

        lID = 0
        sStyle    = "UnReadDataTD"
        lID                 = portal_articles.datasource.recordset.fields("article_id")

        If lID > 0 Then

                lResult    = CCDLookup("id", "portal_articles_read", "user_id=" &
Session("PortalUserID") & " AND article_id = " & lID & "", DBInternet)

                If lResult > 0 Then sStyle = "ReadDataTD"

        End If

        portal_articles.lblStyle.value          = sStyle
```

the record.

What happens is that when the user clicks on the title of the article from the articles grid, the portal_articles_view page is opened.

When the record has finished loading, the Before_Show event performs a call to the UpdateArticlesRead() function.

This function adds the user's id, article_id and date viewed, to the Articles_Read table.

At the top of the article, we need to add another record – this one pointing at the Articles_Read table and displaying the "Archive" tick box.

This code has been commented out, but if you want to allow users to archive the article, remove the apostrophe from the beginning of each line to make the "Archive Form" visible.

If the user places a tick in the "Archive" check box, this record will update the Articles_Read table setting the Archive flag to true.

Notice that we perform a quick check to determine whether there is a record containing both the users id and the article id in the "Portal_Articles_Read" Table.

If it doesn't exist, we call the UpdateArticlesRead() function, otherwise we do nothing since the flag has already been set.

View_approved_articles

```
SELECT
article_id, article_title,
category_id, article_intro,
article_desc, date_begin,
company_id, approved,
display_on_home_page, date_add,
sort_order
FROM
dbo.articles
WHERE
(approved = 1) AND
(display_on_home_page = 1)
```

Controlling The Row Colors Of The Grid

Let's go back to the grid on the "Portal_Articles.ccp" page now and look at the row for the grid.

Notice that we've used a technique discussed in our first issue to enable the dynamic styles using hidden fields.

If you haven't downloaded that issue, you should do so now to fully understand what we're doing here.

We've added some code to the portal_articles_lblStyle_BeforeShow() event for the hidden field called lblStyle and to the portal_articles_lblTitleStyle_BeforeShow() for lblTitleStyle.

This code simply performs a CCDLookUp on the Articles_Read table to determine whether or not a record exists.

If it does, it causes the "ArticleRead" style to be used and if no record is found, a "NewArticle" style is used.

Figure 5 - CSS Styles For Articles Read/Unread

```
<style type="text/css">
<!--
.ReadDataTD{
BACKGROUND: url(http://www.dataobjx.net/images/link_todo.gif) #fff no-repeat left
top;
PADDING-LEFT: 25px;
MARGIN: 0px;
font-family: Verdana,Arial,Tahoma,Helvetica;
font-size: 11px;
font-weight: bold;
COLOR: #000000;
TEXT-ALIGN: left;
BACKGROUND-COLOR: #CCCCCC;
}

.UnReadDataTD
{
BACKGROUND: url(http://www.dataobjx.net/images/link_agenda.gif) #fff no-repeat
left top;
PADDING-LEFT: 25px;
MARGIN: 0px;
font-family: Verdana,Arial,Tahoma,Helvetica;
font-size: 11px;
font-weight: bold;
COLOR: #000000;
TEXT-ALIGN: left;
BACKGROUND-COLOR: cornsilk;
}
-->
</style>
```

Expanding The CodeCharge Studio Portal – Part 1

Let Your Users Know If They've Read The Article Or Not, continued…

Language: .ASP
Database: SQL Server

Modifying The Article_Maint Form

We need to add the new fields that we created to the Article_Maint form.

Add the Intro_Teaser as a TextArea field, add the date fields to indicate when the article should become visible and when it should 'disappear' from view.

And finally, add the "Publish" check box.

Allowing the user to set the date_begins and date_ends fields is a useful feature.

Imagine your going on vacation and want an announcement of some sort to come out while you're relaxing on the beach.

Or perhaps you have a web site that provides the public with the ability to view current events in your area. Having these fields helps to automate the timing of the release of each article.

Creating Our Style Sheets

We're going to use CSS in our style sheet tags used by the articles grid.

The "ArticleRead" tag causes the background color for the cells to appear dim.

Conversely, articles that haven't been read have the "NewArticle" style applied that causes a "New" icon to appear in front of the article title. This style has a brighter background color than the "ArticleRead" style.

Remember, any articles that the user has "Archived" do not appear in the primary grid.

Additional CSS Styles

Take a quick look at the styles we've created for the articles grid further on in this article.

Refer to Figure 5.

These styles include CSS (Cascading Style Sheet) tags to further enrich the display.

Stored Procedures & Views

This application utilizes stored procedures.

We have used stored procedures in order to optimize the speed of code execution.

Naturally, you don't need to use a stored procedure to perform this query, but if you want your application to perform quickly, it's advisable.

Let's have a look at the stored procedures and views.

You might be wondering why we used a cursor based stored procedure for "sa_Portal_Articles_Read**",** since technically – we didn't have to.

At the end of this article, you'll find the other stored procedures and views that are used by the application.

The views, etc. are fairly straight forward but we'll look at a few aspects of each of them.

The views and stored procedures are generally written so as to return only those records that have the 'Publish' flag set to true.

In addition, we need to use a few "Not In" type queries as you'll see in the views and store procedures.

Those of you who are using an MS Access, Oracle or MySQL database for your portal will need to modify some of the 'Queries' a little but the changes you'll need to make relate to largely to the handling of the Date fields within the queries.

You may have noticed that when you click on the title of an article on DataObjx.net, that the color of the article changes to reflect the fact that you've 'read' the article.

By re-creating the data schema and queries for your database, you'll achieve this same functionality.

You Too Can Write An Article For CCS Developer Magazine!

Do you have a special technique you'd like to highlight?

Contact DataObjx Today.

Stored Procedure: sa_Portal_Articles_UnRead

```
Alter Procedure sa_Portal_Articles_UnRead
(
                    @User_ID int,
                    @CatID int
)
As
SELECT *
FROM
  view_portal_articles
WHERE
  view_portal_articles.cat_id = @CatID
AND
  view_portal_articles.article_id not in
(
SELECT
  view_portal_articles_archived.article_id
FROM
  view_portal_articles_archived
WHERE
(view_portal_articles_archived.user_id = @User_ID)
)
ORDER BY date_added DESC, priority_id

Return
```

Expanding The CodeCharge Studio Portal – Part 1

Let Your Users Know If They've Read The Article Or Not, continued…

Language: .ASP
Database: SQL Server

Stored Procedure: sa_Portal_Articles_Read

```
Create Procedure sa_Portal_Articles_Read

(

                    @User_ID int

)

As

SELECT

                    view_portal_articles.article_id, view_portal_articles.article_title, view_portal_articles.article_teaser,

                    view_portal_articles.category_name, view_portal_articles.priority_name

FROM dbo.portal_articles_read

RIGHT OUTER JOIN

view_portal_articles

ON

    portal_articles_read.article_id = view_portal_articles.article_id

WHERE

   (portal_articles_read.user_id = @User_ID)

Return
```

Figure 6 – Function portal_articles_BeforeShow()

```
Dim SQL
Dim Connection
Dim ErrorMessage
Dim lID
Dim sResult
Dim ccSQL

          SQL       = ""
          ccSQL     = ""
          lID       = Request.Querystring("article_id")

          If Len(lID) = 0 Then Response.Redirect "Portal_Articles.asp"

          ccSQL     = "user_id=" & Session("PortalUserID") & " AND article_id = " & lID & ""

          '//First, let's see whether the user has already accessed the article.

          sResult   = CCDLookup("id", "portal_articles_read", ccSQL, DBInternet)

          '// If they haven't read the article already... add a record to the portal_articles_read table.

          If Len(sResult) = 0 Then

                    SQL = SQL & "INSERT INTO portal_articles_read (user_id,article_id,date_accessed) VALUES ("

                    SQL = SQL & Session("PortalUserID") & "," & lID & ",'" & FormatDateTime(Now(), vbShortDate) & "')"
          End If

          Set Connection = New clsDBInternet
          Connection.Open
          Connection.Execute(SQL)
          ErrorMessage = CCProcessError(Connection)
          Connection.Close
          Set Connection = Nothing
          On Error Goto 0
```

Expanding The CodeCharge Studio Portal – Part 1

Let Your Users Know If They've Read The Article Or Not, continued…

Language: .ASP
Database: SQL Server

view_portal_articles_not_archived

```
SELECT
dbo.portal_articles.*,
portal_article_categories.category_name,
Portal_Article_Priorities.priority_name,
Portal_Article_Priorities.sort_order
FROM
portal_articles
INNER JOIN
portal_article_categories
ON
portal_articles.cat_id =
portal_article_categories.category_id
INNER JOIN
Portal_Article_Priorities
ON
portal_articles.priority_id =
Portal_Article_Priorities.id
WHERE
(dbo.portal_articles.date_begins <=
GETDATE()) AND
    (portal_articles.date_ends >=
GETDATE()) AND
    (portal_articles.publish_article = 1)
```

For the most part, those of you using a database other than MS SQL Server the queries that you see here will likely work.

However, you will need to modify any of the SQL that has GETDATE() to the appropriate function for your database.

The SQL Server GETDATE() function returns a Date value as expressed below;

"July 29 1998 2:50 PM"

This is the equivalent of the Visual Basic Now() Function.

Therefore, you will need to locate the appropriate function for the database you're using.

view_portal_articles_archived

```
SELECT dbo.Portal_Articles_Read.*,
dbo.portal_articles.article_title,
    dbo.portal_articles.publish_article
FROM dbo.Portal_Articles_Read INNER
JOIN
    dbo.portal_articles ON
    dbo.Portal_Articles_Read.article_id =
dbo.portal_articles.article_id
WHERE
(dbo.Portal_Articles_Read.archived = 1)
AND
    (dbo.portal_articles.publish_article =
1)
```

Figure 7 – view_portal_articles

```
SELECT dbo.portal_articles.*,
    dbo.portal_article_categories.category_name,

    dbo.Portal_Article_Priorities.priority_name,

    dbo.Portal_Article_Priorities.sort_order

FROM dbo.portal_articles INNER JOIN

    dbo.portal_article_categories ON

    dbo.portal_articles.cat_id = dbo.portal_article_categories.category_id

INNER JOIN

    dbo.Portal_Article_Priorities

ON
    dbo.portal_articles.priority_id = dbo.Portal_Article_Priorities.id

WHERE
    (dbo.portal_articles.date_begins <= GETDATE())
AND

    (dbo.portal_articles.date_ends >= GETDATE())

AND

    (dbo.portal_articles.publish_article = 1)
```

Figure 8 - Function portal_articles_lblEdit_BeforeShow()

```
Dim lngID
Dim sResult
Dim lngAuthorID
Dim strTitle

LngAuthorID = portal_articles.datasource.recordset.fields("added_by_id")

StrTitle = portal_articles.datasource.recordset.fields("article_title")

SResult = ""

If Len(Session("PortalUserID")) > 0 Then

  If CLng(Session("PortalUserID")) = lngAuthorID Then

    lngID = portal_articles.datasource.recordset.fields("article_id")

    sResult = "<a href=""./portal_articles_maint.asp?article_id=" & lngID &
""""><image border=""0"" src=""http://www.dataobjx.net/images/script.gif""
title=""Edit Your Document""></a>  "

  End If

End If

portal_articles.lblEdit.value = sResult
```

Multiple Style Sheets With Oracle & PHP

Controlling Style Sheets For Multiple Companies

Language: .PHP
Database: Oracle

Written By: Fernando Sibaja-Araya. (fsibaja@gmail.com)

In this article we address the issues relating to the use of multiple themes, and also provide our application with the ability to have multiple "services" or "companies".

We're going to create a Contact Administration System that controls the user profiles for multiple companies.

First we will create the tables using the DDL script showed in the fig.1 & 2 (this is a Oracle script).

Re-create the data schema that you see in Figures 1 & 2 using your preferred database.

We will call our project "DynamicThemes".

Create a login page with the wizard and set the security in the Project Setting as displayed in Figure 3.

With the application builder make the search and edit pages for the CONTACT table. For the page containing the CONTACT grid, don't forget to make the CONTACT_maint file the return page of the login record.

If everything is ok at this point, we should have an application that lets us login, search for contacts and add/modify new contact records.

By the way, don't forget to use the same theme for all the pages.

For this example we choose the "coffee" theme as our "Template Theme", I choose it for two reasons:

- I like the coffee theme, and more importantly
- It uses an image for the background and we will have to use a little trick in order to obtain the right behavior (later in this article).

Now that we've replicated the Data Schema mentioned earlier in this article, it's time to add our custom code.

First we will modify the loginUser function in the Common.php file. We will add two session variables to our enviroment: COMPANY and THEME. We need to define two constants: DEFAULT_COMPANY and DEFAULT_THEME.

Your loginUser function should look like the one shown in Figure 3. *Remember that a modified function won't be regenerated.

Now change the datasource of the contact grid in order to show just the records from the company defined in the session. Add this filter to its datasource: CONTACT.COMPANY = Company session variable.

We will also have to change the CONTACT_maint page and set a custom insert as shown in figure 7.

Now create a new page called HEADER. Add the HEADER in all of the pages we already have, except for the login page.

In the HEADER page create a new record form and delete all the components(just leave the <form> tags). Now insert in the record form a new listbox called COMPANY, set its data source to the COMPANY table with the bound_column set to ID_COMPANY and the text_column set to the NAME column.

Set its control_source_type property to code_expression, and the control_source property to CCGetSession("Company").

Add a hidden field called THEME and set its default value to CCGetSession("THEME").

Now go to the html code and include the script shown in Figure 5.

In the listbox html definition set its onchange method to "changeCompany()", like this:

Figure 1 – DDL Script (Oracle)

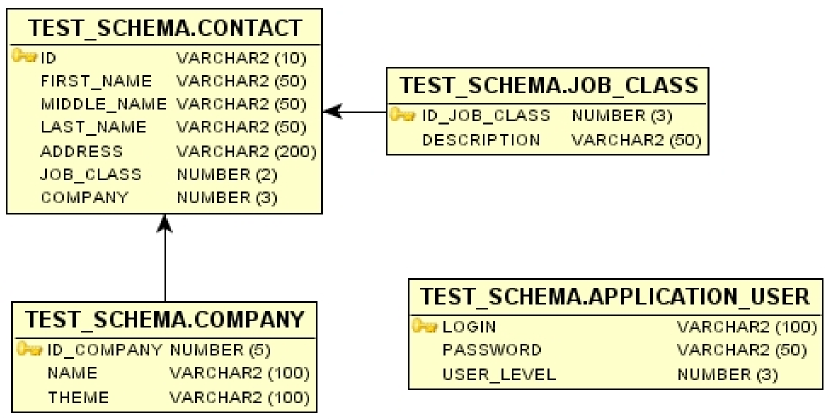


Figure 3 – Login Function

```
define("DEFAULT_COMPANY", 1);
define("DEFAULT_THEME", "Coffee");
...
//CCLoginUser @0-27627269
function CCLoginUser($login, $password)
{
   $db = new clsDBTEST();
   $SQL = "SELECT LOGIN, USER_LEVEL FROM APPLICATION_USER WHERE LOGIN=" .
$db->ToSQL  ($login, ccsText) . " AND PASSWORD=" . $db->ToSQL($password,
ccsText);
   $db->query($SQL);
   $Result = $db->next_record();
   if($Result)
   {
      CCSetSession("UserID", $db->f("LOGIN"));
      CCSetSession("UserLogin", $login);
      CCSetSession("GroupID", $db->f("USER_LEVEL"));
      CCSetSession("Company", DEFAULT_COMPANY);
      CCSetSession("THEME", DEFAULT_THEME);
   }
   $db->close();
   return $Result;
}
//End CCLoginUser
```

Multiple Style Sheets With Oracle & PHP

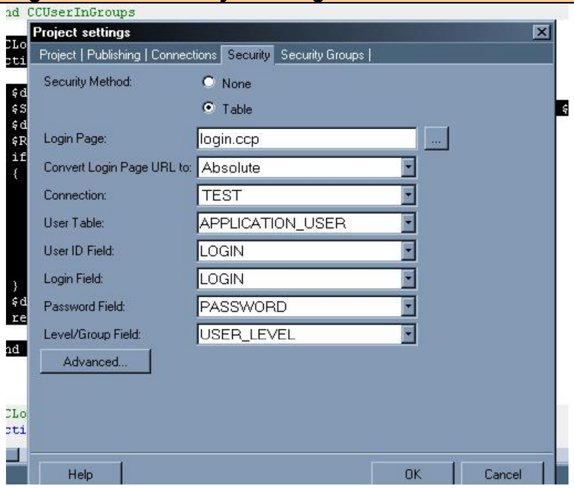
Controlling Style Sheets For Multiple Companies

Language: .PHP
Database: Oracle

Figure 2 – DDL Script (Oracle)

```
--ORACLE  DDL SCRIPT
CREATE USER "TEST_SCHEMA"  PROFILE "DEFAULT" IDENTIFIED BY
    "test" DEFAULT
    TABLESPACE "USERS" TEMPORARY
    TABLESPACE "TEMP" ACCOUNT UNLOCK;
GRANT "CONNECT" TO "TEST_SCHEMA";
GRANT UNLIMITED
    TABLESPACE
    TO "TEST_SCHEMA";

CREATE TABLE "TEST_SCHEMA"."APPLICATION_USER"("LOGIN"
VARCHAR2(
    100) NOT NULL, "PASSWORD" VARCHAR2(50) NOT NULL,
"USER_LEVEL"
    NUMBER(3) NOT NULL,
    CONSTRAINT "PK_LOGIN" PRIMARY KEY("LOGIN");

INSERT INTO "TEST_SCHEMA"."APPLICATION_USER" (LOGIN
,PASSWORD ,USER_LEVEL ) VALUES ('test' ,'test' ,1  )

CREATE TABLE "TEST_SCHEMA"."JOB_CLASS"("ID_JOB_CLASS"
NUMBER(3)
    NOT NULL, "DESCRIPTION" VARCHAR2(50) NOT NULL,
    CONSTRAINT "PK_ID_JOB_CLASS" PRIMARY KEY("ID_JOB_CLASS"));
INSERT INTO "TEST_SCHEMA"."JOB_CLASS" (ID_JOB_CLASS
,DESCRIPTION ) VALUES (1 ,'Web Developer'  );
INSERT INTO "TEST_SCHEMA"."JOB_CLASS" (ID_JOB_CLASS
,DESCRIPTION ) VALUES (2 ,'Sales People'  );
INSERT INTO "TEST_SCHEMA"."JOB_CLASS" (ID_JOB_CLASS
,DESCRIPTION ) VALUES (3 ,'Managment'  );
INSERT INTO "TEST_SCHEMA"."JOB_CLASS" (ID_JOB_CLASS
,DESCRIPTION ) VALUES (4 ,'Instructor'  );

CREATE TABLE "TEST_SCHEMA"."COMPANY"("ID_COMPANY" NUMBER(5)
NOT
    NULL, "NAME" VARCHAR2(100) NOT NULL, "THEME" VARCHAR2(100)
NOT
    NULL,
    CONSTRAINT "PK_ID_COMPANY" PRIMARY KEY("ID_COMPANY"));

INSERT INTO "TEST_SCHEMA"."COMPANY" (ID_COMPANY ,NAME ,THEME
) VALUES (1 ,'DataObjx Consultants' ,'Coffee'  );
INSERT INTO "TEST_SCHEMA"."COMPANY" (ID_COMPANY ,NAME ,THEME
) VALUES (2 ,'English Soccer Academy' ,'Cobalt'  );

CREATE TABLE "TEST_SCHEMA"."CONTACT"("ID" VARCHAR2(10) NOT
NULL,
    "FIRST_NAME" VARCHAR2(50) NOT NULL, "MIDDLE_NAME"
VARCHAR2(50)
    NOT NULL, "LAST_NAME" VARCHAR2(50) NOT NULL, "ADDRESS"
VARCHAR2(
    200) NOT NULL, "JOB_CLASS" NUMBER(2) NOT NULL, "COMPANY"
NUMBER(
    3) NOT NULL,
    CONSTRAINT "FK_ID_COMPANY" FOREIGN KEY("COMPANY")
    REFERENCES "TEST_SCHEMA"."COMPANY"("ID_COMPANY"),
    CONSTRAINT "FK_ID_JOB_CLASS" FOREIGN KEY("JOB_CLASS")
    REFERENCES "TEST_SCHEMA"."JOB_CLASS"("ID_JOB_CLASS"),
    CONSTRAINT "PK_CONTACT" PRIMARY KEY("ID") );
```

Figure 4 – Security Settings


Project settings
Project | Publishing | Connections | Security | Security Groups |
Advanced...
Help
OK
Cancel

Security Method: None [✓ Table]
Login Page: login.ccp
Convert Login Page URL to: Absolute
Connection: TEST
User Table: APPLICATION_USER
User ID Field: LOGIN
Login Field: LOGIN
Password Field: PASSWORD
Level/Group Field: USER_LEVEL

Figure 5 – Function ChangeCompany()

```
<script language="Javascript">

function changeCompany(){

    // -------------------------
    var lbox = document.forms[0].COMPANY;

     if(lbox.selectedIndex!=0){

         var arr = location.href.split("?");
          location.href= arr[0]+"?chserv="+
lbox.options[lbox.selectedIndex].value;

     }
    // -------------------------
}
</script>
```

<select class="CoffeeSelect" name="{COMPANY_Name}" onchange="changeCompany();">

And beneath the hidden field that we have made add a dynamic link to the session theme like this:

<input type="hidden" value="{THEME}" name="{THEME_Name}">
<link href="Themes/{THEME}/Style.css" type="text/css" rel="stylesheet">

Now add a After_initialize event to the HEADER page and add the code shown in Figure 6.

What we want to do is that when the listbox changes, then the page will reload and the session company will change.

Now with all the session variables and code in place, we can change the application's appearance with the 'THEME' session variable.

Again, go to the common files and open the

Multiple Style Sheets With Oracle & PHP

Controlling Style Sheets For Multiple Companies

Language: .PHP
Database: Oracle

template.php file. We will modify the **globalparse** function which is located on or near line **199**.

Instead of the line

echo $this->globals[$block_name],

We will write

echo str_replace (" class=\"Coffee", " class=\"".CCGetSession("THEME"), $this->global.

Remember that we choose "Coffee" as our template theme.

```
  //modified to change the theme on the fly:
  if($output)
    echo str_replace (
"class=\"Coffee" ,
"class=\"".CCGetSession("THEME"),
                    $this->globals[$block_name]);
```

In the database script that we showed before, we created two companies:

- DataObjx Consultants with the coffee theme, and
- English Soccer Academy with the cobalt theme.

As mentioned earlier, we've chosen the coffee theme because it uses a background image. By default, code charge sets the background in the html file like this:

<body ... background="Themes/Coffee/Background.gif" ..>

We need to modify the way the background image is indicated in the <body> tag, as the default is not going to work for the behavior that we want to achieve.

We have to make two changes;

- delete the background reference from the all of the html and
- modify the style.css of the coffee theme like this:

.CoffeePageBODY{ font-family: Tahoma, Verdana, Arial, Helvetica; font-size: 13px; background:url(background.gif); }

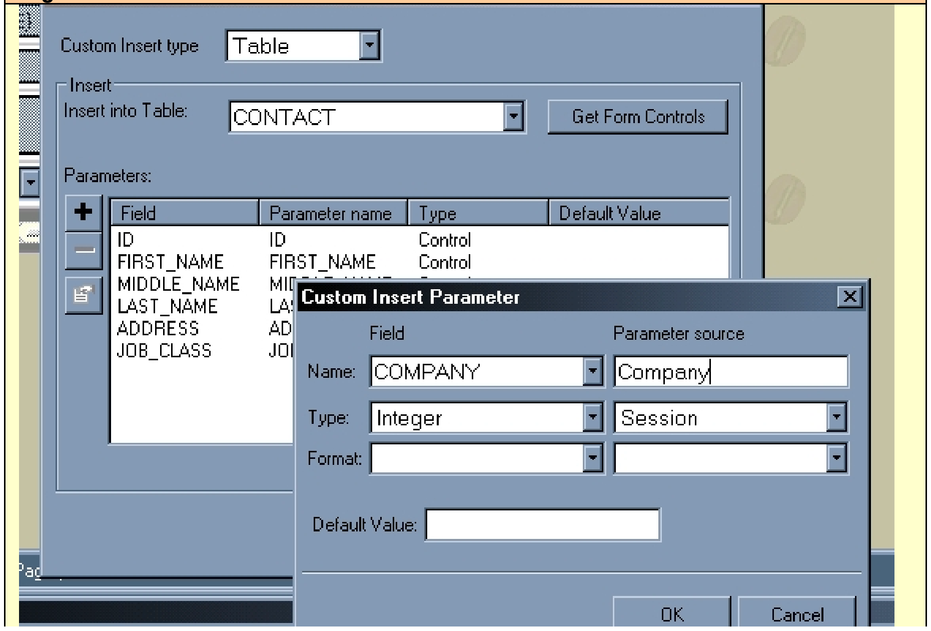
Now don't forget to copy all the themes (the whole directories) that you want to use in the publishing directory.

And that's it!

Figure 6 – Function Header_After_Initialize()

```
//HEADER_AfterInitialize @1-A881B9A6
function HEADER_AfterInitialize()
{
   $HEADER_AfterInitialize = true;
//End HEADER_AfterInitialize

//Custom Code @9-FF1CB783
// -------------------------


 if( $comp = (int)CCGetFromGet("chserv") ){

    if($comp == (int)CCGetSession("Company"))  return;

    $DBoradb = new clsDBTEST();

    $query = "SELECT THEME, NAME FROM  COMPANY WHERE ID_COMPANY =
$comp ";

    $DBoradb->query($query);

    if($DBoradb->next_record()){

            CCSetSession("Company", $comp);
       CCSetSession("THEME", $DBoradb->f("THEME"));

     }

    $DBoradb->close();

  }


// -------------------------
//End Custom Code

//Close HEADER_AfterInitialize @1-F0FCE904
   return $HEADER_AfterInitialize;
}
//End Close HEADER_AfterInitialize
```

Figure 7



Custom Insert type: Table

Insert
Insert into Table: CONTACT
Get Form Controls

Parameters:

Field	Parameter name	Type	Default Value
ID	ID	Control	
FIRST_NAME	FIRST_NAME	Control	
MIDDLE_NAME	MI		
LAST_NAME	LA		
ADDRESS	AD		
JOB_CLASS	JO		

Custom Insert Parameter

	Field	Parameter source
Name:	COMPANY	Company
Type:	Integer	Session
Format:		

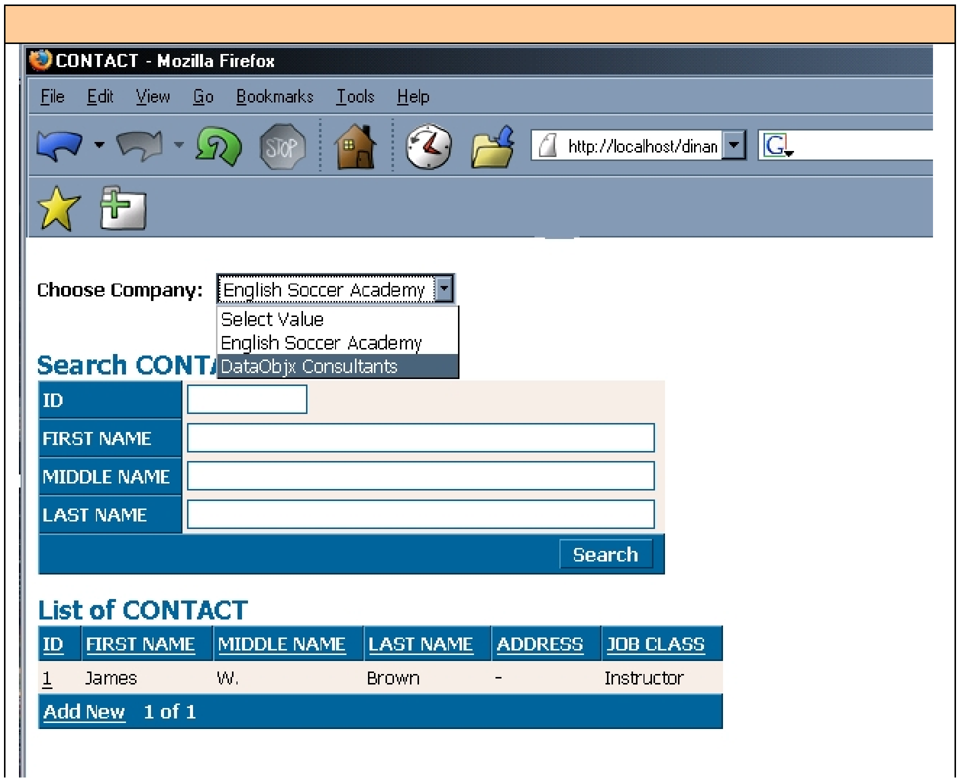
Default Value:

OK Cancel

Multiple Style Sheets With Oracle & PHP

Controlling Style Sheets For Multiple Companies

Language: .PHP
Database: Oracle


CONTACT - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/dinan

Choose Company: English Soccer Academy

Select Value
English Soccer Academy
DataObjx Consultants

Search CONTACT

ID	
FIRST NAME	
MIDDLE NAME	
LAST NAME	

Search

List of CONTACT

ID	FIRST NAME	MIDDLE NAME	LAST NAME	ADDRESS	JOB CLASS
1	James	W.	Brown	-	Instructor

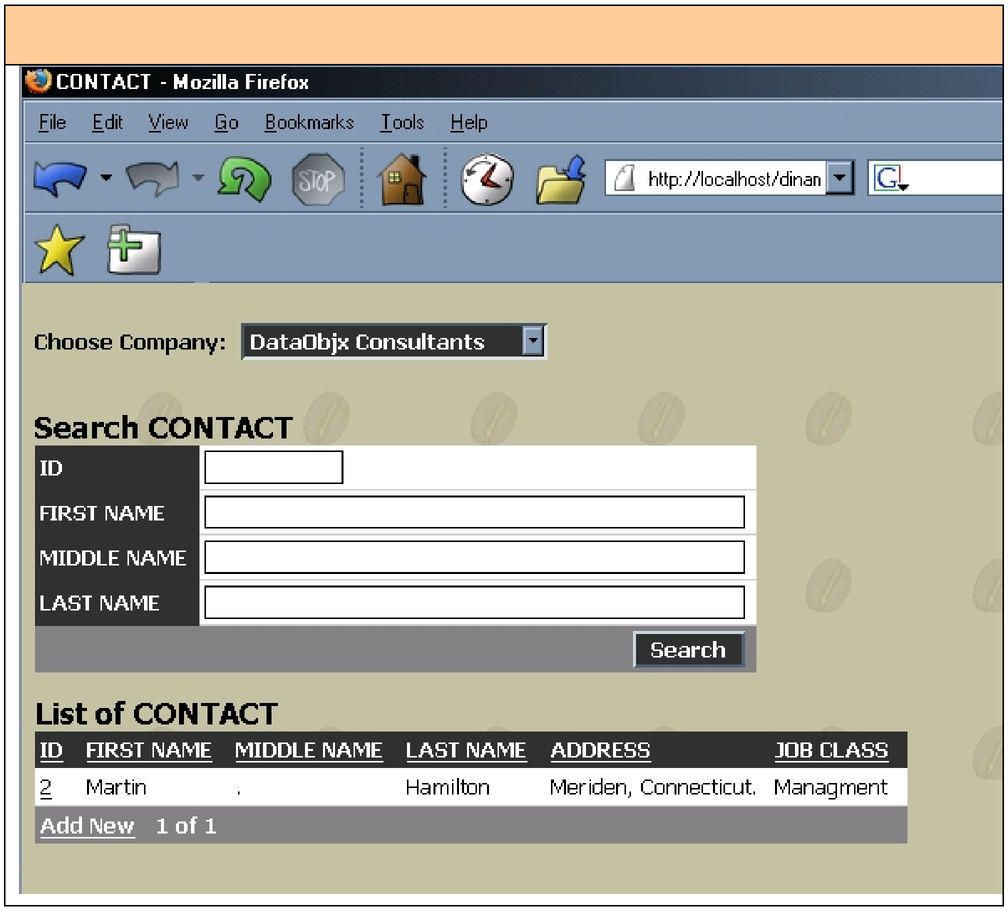
Add New 1 of 1

We have an application suitable for many companies using many themes.

About Fernando Sibaja-Araya

Mr. Sibaja is a graduate computer engineer from ITCR (Instituto Techologico de Costa Rica), and develops data driven web applications using CodeCharge Studio, PHP, oracle and ProgressSQL.

Costa Rica


CONTACT - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/dinan

Choose Company: DataObjx Consultants

Search CONTACT

ID	
FIRST NAME	
MIDDLE NAME	
LAST NAME	

Search

List of CONTACT

ID	FIRST NAME	MIDDLE NAME	LAST NAME	ADDRESS	JOB CLASS
2	Martin	.	Hamilton	Meriden, Connecticut.	Managment

Add New 1 of 1

Building A Document Management System

FEATURED ARTICLE: Part 2 of a three part series…

Language: .ASP
Data Base: MS Access

Document Manager Tables Description
Users* Table Needed to ascertain the login parameters and the security level for a given user.
DocumentManagerDepartments Table Used to hold the names for various 'area's' or departments into which a user can assign a document.
DocumentManagerUpload Table Used to hold the binary files and information relating to the binary file.
DocumentManager_Upload_To_Users Table Used to hold the values necessary to assign documents to particular individuals.
* Note: The Users table used in this example is a cut-down version of the MS Access files that come in the CodeCharge Studio Example Pack. Therefore, when you upload the tables to SQL Server, MySQL or Oracle, etc – ensure that you do not overwrite *your* Users table.

In our previous issue, we began building our document management system from the ground up.

We created the primary tables that we needed to store the files that we uploaded to the application and we created a wizard interface to make the upload process easy to the end-user.

In this issue, we're going to show you how to allow your users to download the files from the database and we'll show you how to count the number of times the file was downloaded.

In addition, we're going to begin to incorporate our dynamic security schema to allow the 'owner' of the file to stipulate the actions each user can take for any given file. In our next issue, we're going to show you how to provide 'check-in/check-out' capabilities. You may wonder why we didn't show you how to do this in this issue, but as you will see – what we're showing you in this issue is fairly complex – so we wanted to ensure that we don't get too far ahead of ourselves.

Therefore, in this issue we're going to focus on the primary issues associated with assigning files to specific users - And as you will see – this is a complex task requiring code that is – well, more complex than you might initially realize.

By the end of this article, the application will contain enough functionality for you to get a fairly sophisticated application up and running – but we're not done yet.

In fact, our next issue will take the existing code even further – allowing your users to 'Check-In and Check-Out' documents.

Let's Begin

If you refer to our previous issue (page 12), you'll notice that we made reference to a table called '**DocumentManagerUploads2Users**'.

We've changed this to "**DocumentManager_Upload_To_Users**" in an effort to be more linguistic. E.g., although two (2) and 'To' has the same sound in English – that's not true for all languages.

The name we've given to this table may have provided you with a clue as to how we will assign files to individual users, and you'd be right.

But if that was all it was for, it'd only be half of the story. This table also dictates what actions a user can take and in our next article, we'll be expanding on the actions a given user can take for a given document. This is one method for providing 'dynamic' security to a file or page and the technique demonstrated can be extrapolated upon to achieve these capabilities.

In this issue, we're going to provide the end user with relatively basic capabilities.

These capabilities are;

- The ability to download the file/document (default), and
- The ability to edit/overwrite the file/document

Enough said. Let's get started.

Configuration

Before we begin, let's take a quick look at a new file called "ConnectionString.asp".

This file contains the connection string information required to connect your application to your database.

Rather than have connection string code contained within a half dozen pages, we're going to place the connection string information inside of one file and this file will be called as an 'include' from many files.

This will make it much easier for you to configure the application.

Open the "ConnectionString.asp" file and have a look at the code.

Modify the code to allow your application to connect to the database that you're using for the Document Management System.

Allowing The User To Download

Document Management System Part 1	Document Management System Part 2	Document Management System Part 3
In our previous article we created the following features; • Address Methodology • Create the database tables • Create the reusable 'Upload Wizard' • Incorporate upload capability, storing the binary file to the database table.	In this issue, we will; • Allow users to download the files from the database • Count the number of times the file was downloaded • Define what rights a user has for a given document. • Refine the 'Upload Wizard' • Amend the menu, etc.	In our next issue, we will; • Create the form to allow users to allocate documents to specific users. • Allow the owner of the document to control the actions users can take with the document including 'check-in' and 'check-out' capability.

Building A Document Management System

Language: .ASP

FEATURED ARTICLE: Part 2 of a three part series…

The File & Incrementing The Number Of Times Downloaded

In our previous article, we integrated the PureASPUpload component to allow users to upload a file directly into a database field.

This was good, because it allowed you to provide your users with the ability to upload certain files that would otherwise be a cause of concern. In other words, users could now upload executable files or other files that could have compromised the security of your web site.

Now, we're going to show you how to allow users to 'download' files. In addition, we're going to provide a mechanism that will track the number of times the file was downloaded.

In order to do this in such a way that we could move forward – step by step, we've separated this process into two (2) pages.

The first page, called "incrementdownload.asp" (figure 2 and figure 3) records the number of times the 'download' icon was clicked on.

Open that page now and have a look at the code.

The purpose of this page is simple. Its job is to increment the field that contains a count of the number of times the file was downloaded.

That's all this file does. It accesses the database, gets the number of times the file was downloaded "times_downloaded" field – increments the number by one (1) and writes the new value back to the "DocumentManager_Upload" table.

When this is done, the code redirects to the "db_down.asp" page which performs the actual downloading of the file.

The user will get the normal download or 'open' capability, but we won't allow the user to 'open' the file – only download it.

The reason for this is because when we incorporate the "Check-In/Check-Out" capability – we need to ensure that the file was checked out and not simply opened.

The download code (db_down.asp) is a cut down version of the code supplied when you download the PureASPUpload code.

Associating A File With A User

Before we can allow a user to have access to a file, we need to ensure that the user *has* access to the file.

To do this we will utilize a somewhat complex bit of code contained within the page called "**DocumentManager_FileToUsers.asp**".

This code differs somewhat from the code used in the CodeCharge Studio tutorial demonstrating a similar technique [**CCS Examples: Updating Many-to-Many via Two ListBoxes**]. There's nothing at all wrong with the tutorial, it just isn't suitable for our purposes.

How? Well, the CCS tutorial demonstrates a method involving the deletion of *all* records associated with a file and then the code re-inserts the records causing a new primary key [@@Identity/Autonumber ID] to be created.

If we used this technique in our code, the file association would be disrupted, thus causing orphaned records or 'lost primary keys' associations with another table.

In addition, if we left it to the methodology shown in the tutorial, we'd have to re-assign – each time – the users 'rights' for any given file because the original record is destroyed an then re-inserted.

This won't work for our needs, so we need to perform a more complex technique that will maintain the rights we've already assigned without changing the Upload_ID.

We also want to create a more efficient way of dealing with the inserts and updates to the DocumentManager_Upload table.

Just in case you haven't attempted to replicate the code demonstrated in the CCS Tutorial and have become lost, here's what happens in the tutorial code.

When the user first assigns users from the "Available List Box" to the "Assigned List Box" a record (Insert - invoking a new AutoNumber/Identity ID) is created.

If you subsequently "Add" another user what's occurring is actually an "Update" and consequently all users are deleted and then re-inserted along with the new record. Thus all original ID's are given a new Autonumber/Identity ID…. Because they were all deleted and then inserted again.

The problem arises when you have a third table relying on that AutoNumber/Identity ID field. Since the record is deleted and re-inserted gaining a new ID, the record in the third table is no longer valid. To correct it, you'd need to write code to 'Update' the third table to account for the new Autonumber/Identity ID caused by the update.

This is not as efficient as the method we discuss – but, again we have a more complex task ahead of us.

Many To Many ListBox Associations

What we're going to do is more complex – but – it doesn't break third table ID associations when we *update* the Many-To-Many Listboxes.

The concept is simple and logical but the implementation is deceivingly complex.

What we want is this;

- Add the new record
- Determine whether anyone in the "Assigned List Box" has been removed and delete them,
- Mean-while – leave the rest of the original records intact.

The difficulty associated with this somewhat simple logic is not as easy as it sounds however.

Have a look at the code in the 'events' page.

We need to get the original records already located in the "Assigned" list box (figure 1), then we need to determine what users have been removed from that list box so we can delete those records and then we need to add the new user id's transferred to the "Assigned" list box.

Refer to the function called: **Function ModifyFileAssignment(Actions)**

These Many-To-Many list boxes are great for the end user – but they're

Building A Document Management System

Part 1 of a three part series, continued…

real time burners for programmers.

As a word of caution… any time you're required to write a Many-To-Many list box form… give yourself a couple of additional *hours* for the task.

Additional Wizard Pages

In the first article we built our "Select_Update_Action.asp" to allow the end-user to perform update actions against various segments of the uploaded record and we now need to add another one.

Open this page and notice that we've added a link called "Assign File To Public Folder Or Specific Users".

When this link is clicked we need to provide the end-user with the ability to perform one of the two (2) possible actions.

Figure 1

```
Function GetOriginalRightHandListBoxRecords(vUploadID, vReturnValue)
Dim SQL
Dim cn
Dim rs
Dim pkID
Dim sFunctionName
  '... used by error message
  sFunctionName="GetOriginalRightHandListBoxRecords()"

          If Len(vUploadID) = 0 Then
                    Response.write "vUploadID Not Passed to " & sFunctionName
                    Response.end
          End If

          SQL = ""
          SQL = SQL & "SELECT DISTINCT user_id "
          SQL = SQL & "FROM DocumentManager_Upload_To_Users "
          SQL = SQL & "WHERE (upload_id = " & vUploadID & ") "
          SQL = SQL & "ORDER BY user_id "

          GetOriginalRightHandListBoxRecords = False  'Assume Failure

  'Create a new database connection object
  'replace clsDBDocManager with the class name for your projects connection
  Set cn=New clsDBDocManager
  cn.Open

  Set rs = cn.Execute(SQL)

  If cn.Errors.Count = 0 Then
    If NOT rs.EOF Then
                    While Not rs.EOF
                    vReturnValue = vReturnValue & CCGetValue(rs, "user_id") & ","
                              rs.movenext
                    Wend
                    If Mid(vReturnValue, Len(vReturnValue),1) = ","  Then
                              vReturnValue = Mid(vReturnValue, 1, Len(vReturnValue)-1)
                    End If
                    GetOriginalRightHandListBoxRecords = True
          Else
                    Exit Function
    End If

    'Close the recordset
    rs.Close
    Set rs = Nothing
  Else
            Response.Write "<font color=""red""><b>WARNING! </b></font><font color=""darkgreen""><b>" & sFunctionName &
"</b></font> returned errors..."
      cn.Errors.Clear
            Response.End
  End If

          'Close and destroy the database connection object
          cn.Close
          Set cn = Nothing
End Function
```

Building A Document Management System

Part 1 of a three part series, continued…

We have therefore created the "DocumentManager_AssignFile.asp" page to channel the user into the desired action screen.

The previous article showed how to assign the file to a "Public" folder. The "DocumentManager_FileToUsers.asp" page allows the end-user to assign the file to 'specific users'. This is the page containing the Many-To-Many list boxes.

We need to account for the possibility that the file was originally assigned to a public folder and subsequently may get assigned to specific user(s). If this occurs we need to remove the "Public Folder" associations. A file is either available to everyone or to specific users… not both.

We could further expand the capability to allow for the file to be assigned to a specific group plus certain users – but that is outside the scope of this particular article.

If the file is deleted we need to delete all file associations for the file in the "DocumentManager_Upload_To_Users.asp" table - so that the table doesn't end up with orphaned and un-needed records.

Enough said about this seemingly simple but relatively complex task. Since this is some rather lengthy code, refer to the events page for a more complete understanding of how this is programmatically accomplished.

Applying Dynamic Security To The File

In the third part of this series, we'll be expanding the 'edit/overwrite' capability to include "Check-In" and "Check-Out" capabilities and, well - we will leave that for the next issue.

For now, let's have a look at the "DocumentManager_Upload_To_Users" table.

Take notice of the user_id field and the owner_id field. The user_id field is the id of the user given access to the file and the owner_id is the id of the user that uploaded the file to begin with.

Also take notice of the

- Can_edit,
- Can_delete, and
- Can_download fields.

These are the fields (bit flags) that we'll use to 'dynamically' allocate what actions a user can and cannot perform with any given file/article.

The 'Can_Download' Field may make you consider the following – "If I didn't want the user to be able to download the file, I simply wouldn't have associated the file to the users – user_id. But as you'll see in the next issue – the ability to download the file vs. the ability to "Check-In" and "Check-Out" the file are two (2) different things.

You can use the method demonstrated here to create dynamic security or 'access' to a file or a web page, etc.

Figure 2

```
Function GetTimesDownloaded()
Dim SQL
Dim rs
Dim cn
Dim lngValue

      If Len(ID) = 0 Then
            response.write "No ID"
            Response.end
            'Exit Function
      End If

  Set cn = Server.CreateObject("ADODB.Connection")
  Set rs = Server.CreateObject("ADODB.Recordset")
  cn.ConnectionString = GetConnection
  cn.Open

      SQL = "select times_downloaded from DocumentManagerUpload where UploadID=" & ID
   Set rs= cn.Execute(SQL)

      If NOT rs.EOF Then
            If IsNull(rs("times_downloaded")) Then
                  lngValue = 0
            Else
                  lngValue = rs("times_downloaded")
            End If
      Else
            lngValue = 0
      End If
      GetTimesDownloaded  = lngValue
    rs.Close
    Set rs= Nothing
    cn.Close
    Set cn = Nothing
End Function
```

Building A Document Management System

Part 1 of a three part series, continued…

The concept is the same although the fields may be different for your application if you're applying the concept to a web page.

If you were applying this technique to perform dynamic security for web pages, you'd start by adding another field to this table called perhaps 'page_name' and go from there.

We *will* run an article on applying dynamic capability for accessing pages on you web site in a future article. But as you can see, this is a somewhat complex issue and we need to remain focused on the document management system for now.

Essentially, what we've done is create a table to associate the users user_id with the upload_id of the file.

If the association exists, the user can perform one or more actions against the file.

What's Next

In our next article, we'll be adding "Check-In/Check-Out" capability to our document management system.

Adding "Check-In/Check-Out" capability is some fairly complex code as well and since the techniques demonstrated in this issue were fairly involved, we wanted to make sure that you do not get lost along the way.

In addition, we'll begin to tighten up the code and security on each page to close any loop holes that might exist.

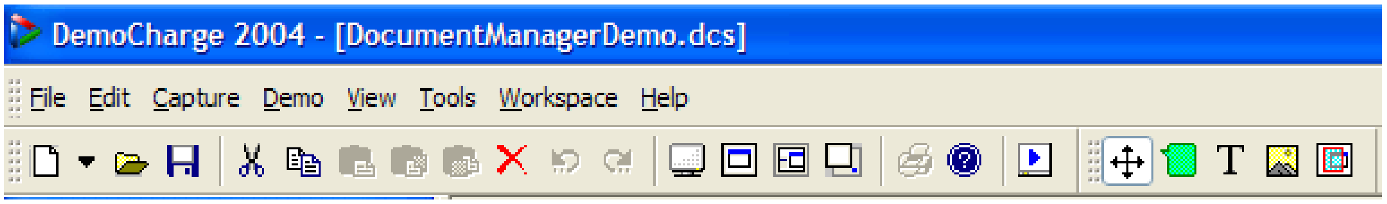
We'll add code to the db_down.asp page to ensure that the user has sufficient rights to access the file and prevent an action against a file if a user has attempted to modify the upload_id passed in by the query string.

Figure 3

```
Function IncrementTimesDownloaded()
Dim intRecordsAffected
Dim cn                          '// Database connection
Dim cmd                          '// Command object for stored procedure
Dim prm                          '// Parameter object for stored procedure
Dim SQL
Dim TimesDownloaded

  Set cn = Server.CreateObject("ADODB.Connection")
  Set cmd = Server.CreateObject("ADODB.Command")

  TimesDownloaded = GetTimesDownloaded()
  If Len(TimesDownloaded) = 0 Then TimesDownloaded = 0
  TimesDownloaded = TimesDownloaded + 1

  SQL = "Update DocumentManagerUpload Set times_downloaded = " & TimesDownloaded
  SQL = SQL & " where UploadID=" & ID

'response.write SQL & "<BR>"
'response.end

  cn.ConnectionString = GetConnection
  Set cn = GetConnection
 ' cn.Open

  Set cmd.ActiveConnection = cn
  cmd.CommandType = adCmdText
  cmd.CommandText = SQL

  cmd.Execute intRecordsAffected

If intRecordsAffected > 0 Then
  '// Return Id of record just added - the stored procedure automatically returns this
  IncrementTimesDownloaded= True  '//cmd.Parameters("@ID").Value
Else
     Response.write Err.Number & "<BR>" & Err.Description
     REsponse.end
End If

  Set cmd = Nothing
  Set prm = Nothing
  cn.Close
  Set cn = Nothing

End Function
```

DemoCharge Standard Edition

A Short Review…


DemoCharge 2004 - [DocumentManagerDemo.dcs]
File Edit Capture Demo View Tools Workspace Help

YesSoftware has just released their new product called DemoCharge.

This product is innovative, powerful and slick.

This application allows you to capture and record mouse movements, keystrokes and so forth and save them to a .gif or .avi file (standard version).

At first glance, you may look on the application as a great way to 'demo' a product – and you'd be right.

But if you think of it in that way and go no further, you'd be doing yourself a disservice.

This application can be used for much more than just demo's of your product.

The Cost Of Training And Other Opportunities

One of the things that cost organizations quite a large sum of money is training.

People come and go, move to other departments, get sick or go on vacation.

If you look around you while you're at work, you'll see a thousand ways that you can benefit the organization using DemoCharge.

Simply pick one department and look at some of the documents they have to fill out. I'm not talking about the simple vacation forms either. Pick one of the forms that has to be filled out for a particular purpose – one that only one or two employees have to use. Some of these forms can be relatively complex and others require a certain expertise or technical skills.

What you'll see is an opportunity waiting to happen.

Imagine providing your organization with in-house 'self paced' training courses.

Or better yet, perhaps you might see an opportunity to create a new business line of your own… by creating training courses for many organizations!

In fact, by using the DataObjx Document Manager combined with DemoCharge – you can indeed create a web based training facility for clients.

Something to consider!

Using DemoCharge, you can empower an organization in ways that they thought would cost thousands of dollars.

By creating "how-to's" you can empower a department and the employees within the organization!

Let's take a look at DemoCharge by creating a simple tutorial.

Since CCS Magazine has been creating the Document Management System – we'll use this for our example.

Before We Begin

Before we begin to create our tutorial, you need to consider the resolution of the end users computer.

As developers we often have our screen resolution set at 1024x768.

When we began to create our demo's – we created and viewed them at this resolution.

However, when we sent a demo to one of our clients they reported that the demo looked 'scrunched' or fuzzy. After some enquiry we realized that the client had their monitor resolution set at 800x600.

This gave us the clue we needed and we recreated the demo after setting our screen resolution to 800x600 as well.

After we re-recorded the demo (now at 800x600) and sent it to the client – they reported that the demo was now crisp and clear.

Afterwards, we realized that when you go to create the .gif file that you can specify the output size as shown in figure 1 - And that we didn't have to reset the resolution of our own monitors.

However, it demonstrated an important point. If the end user has their resolution set at 800x600 you need to set the output size to "Custom" and then specify a width of 800 and height of 600 for your output… or they will see the .gif as being 'fuzzy'.

Conversely, if you've created the .gif using the 'custom 800x600' output and the end user has their monitor set to a higher resolution – it's simply bigger.

So, depending on your audience – you may want to make a habit of setting your .gif to be generated at 800x600 as a default.

That way, everyone is able to view the demo without issue.

Lets Begin

Admittedly, it's difficult to perform a review on a product that's capable of 'reviewing itself'. YesSoftware did a great job of placing animated .gifs within the help file and it's an impressive start considering that the application is so new.

This has made it somewhat difficult to perform a 'review as such – so rather than repeat what you'll find in the help file, we thought you might be better off if we provided you with some ideas as to how you can profit from the utilization of the application.

Entrepreneurial Considerations

We began by creating a small demo, adding 'balloon' comments and compiling the demo into an .avi file.

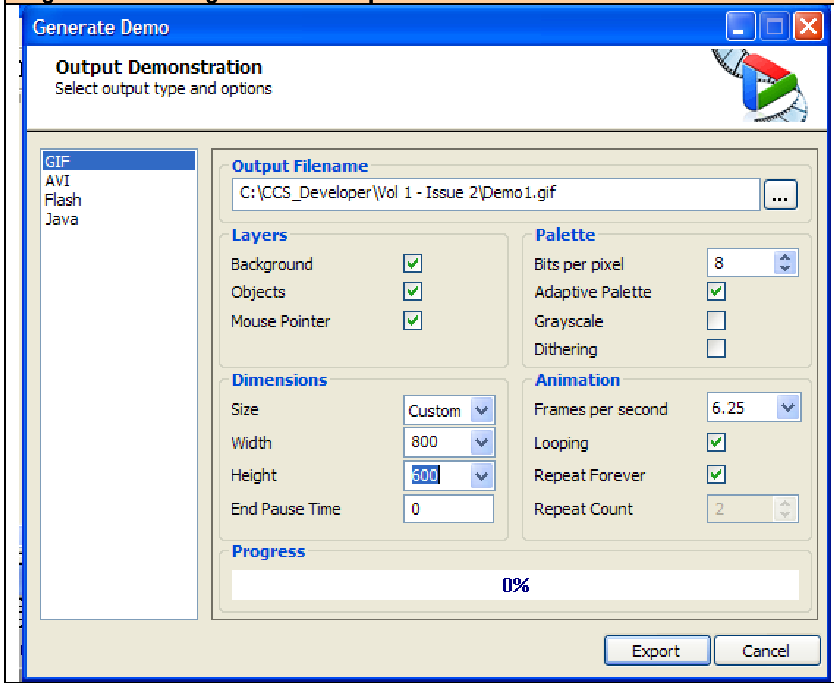
The .avi file was indeed very large considering that the demo was 'short and sweet', but as mentioned in the DemoCharge Help file – there are third party applications that can bring this size down considerably.

After creating the demo we began canvassing managers in one of our

Building A Document Management System

Part 1 of a three part series, continued…

Figure 1: Setting Custom Output


Generate Demo

Output Demonstration
Select output type and options

GIF
AVI
Flash
Java

Output Filename
C:\CCS_Developer\Vol 1 - Issue 2\Demo1.gif

Layers	
Background	✓
Objects	✓
Mouse Pointer	✓

Palette	
Bits per pixel	8
Adaptive Palette	✓
Grayscale	
Dithering	

Dimensions	
Size	Custom
Width	800
Height	600
End Pause Time	0

Animation	
Frames per second	6.25
Looping	✓
Repeat Forever	✓
Repeat Count	2

Progress
0%

client organizations by showing them the demo and asking them to consider the forms and applications used by various staff within their departments.

We asked them to think about those applications/forms that required a certain expertise to use or complete and especially to consider those applications and forms that were used in areas with a high turn-over… e.g., managers were continually engaged in costly training exercises.

As we watched the clients – who were watching the Demo, we could see the wheels begin to spin. They were clearly thinking about the benefits.

Have a look around you after trying out DemoCharge. Opportunity is everywhere!

Since DemoCharge came out so close to the release of this issue, we need more time to investigate the in's and out's of the application.

So, we plan to continue this review in future issues and look forward to commenting on the Professional version when it's released.

Watch for more articles on this incredible application that opens up opportunities for you!

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!