

# EEP JQuery 進階課程

訊光科技 / Andy Kao





## 進階課程大綱

進度	課程內容	時間
上午	Format控制	70
	SetWhere控制	
	下課休息	20
	DataForm控制	70
下午	DataGrid控制	
	中午休息	12 : 00-13 : 15
	元件補充設計	70
	下課休息	15
	其他設計	70

INFO LIGHT

## Format控制

INFO LIGHT

## HyperLink

ID	NAME	CITY
10220	JOHN	
10221	JOHN	COTT
10253	JOHN	BOITM
10234	JOHN	ECONEX
10255	JOHN	ELIAS

WORDERS上的CustomerID欄位

```
<script>
function LinkScript(val, rowData)
{
    return '<a href="WCUSTOMERS.aspx">' + val + '</a>';
}
</script>
```

INFO LIGHT

補充

## HyperLink

- ◆ 另開一個Tab分頁並傳遞參數到新頁面

```
function LinkScript(val, rowData) {  
    return "<a href='javascript: void(0)' onclick='LinkReply(\"" + val + "\" );'>" + val + "</a>";  
}  
function LinkReply(val) {  
    parent.addTab("客戶管理", "CLASS0912/WCUSTOMERS.aspx?CustomerID=" + val);  
}
```

- ◆ 新頁面接收參數並做過濾

```
<script>  
var CustomerID = Request.QueryStringByName2("CustomerID");  
var flag = true;  
function dgOnLoadSuccess() {  
    if (flag) {  
        $("#dataGridView").datagrid('setWhere', "CustomerID='" + CustomerID + "'");  
    }  
    flag = false;  
}
```

INFOLIGHT

---

---

---

---

---

---

---

---

## ToolTips



WORDERS上的ShipAddress欄位

```
function genToolTip(val, rowData) {  
    if (rowData.ShipAddress != undefined) {  
        return "<p title='" + rowData.ShipAddress + "'>" + val + "</p>";  
    }  
    else {  
        return val;  
    }  
}
```

INFOLIGHT

---

---

---

---

---

---

---

---

## ShowMemo



WORDERS上的ShipName欄位

```
function genShowMemo(val, rowData) {  
    return '<textarea rows="3" disabled="disabled">' + val + '</textarea>';  
}
```

INFOLIGHT

---

---

---

---

---

---

---

---

## Checkbox

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	Discontinued
1	Chai	1	1	10 units/bottle	10	0
2	Clam	1	1	10 units/bottle	10	0
3	Coriander Oil	1	2	10 units/bottle	10	0
4	Coriander Oil	1	2	10 units/bottle	10	0
5	Coriander Oil	1	2	10 units/bottle	10	0
6	Coriander Oil	1	2	10 units/bottle	10	0
7	Coriander Oil	1	2	10 units/bottle	10	0
8	Coriander Oil	1	2	10 units/bottle	10	0
9	Coriander Oil	1	2	10 units/bottle	10	0
10	Coriander Oil	1	2	10 units/bottle	10	0

WPRODUCT上的Discontinued欄位

- EditorOptions設定: on:1, off:0
- 支援欄位: Bit(True/False), int(1/0), Char (Y/N)
- 顯示設定: Format : L, 是, 否  
L, checkbox



## 自動折行

客戶名稱	聯絡人	職稱	地址
1	張光利科技系統股份有限公司	ANDY KAO	Accounting Manager
2	張光利科技系統股份有限公司	ANDY KAO	Marketing Manager
3	Alfreda Futterkiste	Maria Anders	Sales Representative

WCUSTOMERS上的Address欄位

```
function wrapText(value) {
    return "<p style='margin:0px;word-wrap:break-word:white-space:
normal'>" + value + "</p>";
}
```



## 自動折行-refval應用

輸入	查詢	地址	城市
1	22	臺北市信義區信義路二	臺北市
2	30000000	臺北市信義區信義路二	臺北市
3	30000000	臺北市信義區信義路二	臺北市
4	30000000	臺北市信義區信義路二	臺北市
5	30000000	臺北市信義區信義路二	臺北市
6	30000000	臺北市信義區信義路二	臺北市
7	30000000	臺北市信義區信義路二	臺北市
8	30000000	臺北市信義區信義路二	臺北市
9	30000000	臺北市信義區信義路二	臺北市
10	30000000	臺北市信義區信義路二	臺北市

解決在GridView中使用  
inforefval對照參考檔後，  
再寫formatScript來換行  
顯示，造成顯示的內容  
又變成value，而不是  
reference後的內容

WCUSTOMERS上的Address欄位

```
function myformat(val, row) {
    var displayValue = eval('getTextField').call(this, val, row);
    return "<p style='margin:0px;word-wrap:break-word:white-space: normal'>" +
displayValue + "</p>";
}
```



## SetWhere控制



---

---

---

---

---

---

---

## DataGrid SetWhere

### •預設Where(WCUSTOMERS)

在dataGridView的OnLoadSuccess屬性中定義一個js方法，  
比如：MyWhere，寫如下程式：

```
var flag = true; //定義一個全域變數，只有第一次執行  
  
function MyWhere() { //執行setwhere方法，過濾自己的條件  
    if (flag) {  
        $("#dataGridView").datagrid('setWhere', "CustomerID like 'A%'");  
        flag = false;  
    }  
}
```



---

---

---

---

---

---

---

## Query控制

### •配合DataGrid的Query條件

WCUSTOMERS頁面上添加js程式

```
function queryGrid(dg) { //查詢後添加固定條件  
    if ($(dg).attr('id') == 'dataGridView')  
    {  
        var where = $(dg).datagrid('getWhere');  
        if (where.length > 0) {  
            where = where + "and CompanyName like 'B%'";  
        }  
        $(dg).datagrid('setWhere', where);  
    }  
}
```



---

---

---

---

---

---

---

## Refval控制

•DataForm上refval元件預設帶入Where條件後第一筆資料在DataForm的OnLoadSuccess裡先過濾資料後，再將過濾後資料的第一筆值帶入元件上(WCUSTOMERS)

```
function DFOnLoadSuccess() {  
    $('#dataFormMasterContactTitle').refval('setWhere',  
    "CompanyName='訊光科技系統股份有限公司'");  
    setFirstValue($('#dataFormMasterContactTitle'),  
    "CompanyName='訊光科技系統股份有限公司'");  
}
```



---

---

---

---

---

---

---

## DataForm上元件setwhere

- JQRefVal 元件：  
\$\$('#id').refval('setWhere',wherestring)；
- JQComboBox 元件：  
\$\$('#id').combobox('setWhere',wherestring)；
- JQComboGrid 元件：  
\$\$('#id').combogrid('setWhere',wherestring)；
- JQOptions 元件：  
OnWhere事件，讓資料可以動態過濾。return完整條件  
function opwhere() {  
 return "Customers.City='Berlin'";  
}



---

---

---

---

---

---

---

## DataGrid上元件setwhere

- JQRefVal元件(WPRODUCT)  
基本上dataGrid本身是不支持WhereItem的WhereField用法的。不過有一種方式可以達成依賴欄位值做過濾的效果。寫在whereItem的whereMethod

```
function RFilter() {  
    var row = $('#dataGridMaster').datagrid('getSelected');  
    var value;  
    if (row != null) {  
        var rowindex = $('#dataGridMaster').datagrid('getRowIndex', row);  
        var ProductName = $('#dataGridMaster').datagrid('getEditor', { index: rowindex, field: 'ProductName' });  
        if (ProductName != null) {  
            value = $(ProductName.target).val();  
        }  
    }  
    return value;  
}
```



---

---

---

---

---

---

---

## DataGrid上元件setwhere

☞ JQComboBox元件、JQComboGrid元件：

OnBeforeLoad屬性上定義方法

```
function comboBeforeLoad(param) {  
    var queryWord = new Object();  
    queryWord.whereString = "ProductName like 'C%'";  
    param.queryWord = $.toJSONString(queryWord);  
}
```



---

---

---

---

---

---

---

## DataForm控制



---

---

---

---

---

---

---

## 檔案上傳與下載

(WCUSTOMERS)

☞ 檔案上傳：  
Infofileupload元件，EditorOptions裡設定屬性UpLoadFolder，  
為上傳的目錄：Files/Upload

☞ 檔案下載：(放在DataGrid欄位的Format屬性上)  
download,Folder:Files/Upload

☞ 動態控制上傳目錄  
function onBeforeUpload(options) {  
 options.upLoadFolder = 'new folder';  
 return true;  
}

GridRow	Download	Picture
18	<input checked="" type="checkbox"/>	img271428471
19	<input checked="" type="checkbox"/>	123456789.jpg
20	<input checked="" type="checkbox"/>	123456789.jpg
21	<input checked="" type="checkbox"/>	123456789.jpg
22	<input checked="" type="checkbox"/>	123456789.jpg
23	<input checked="" type="checkbox"/>	123456789.jpg
24	<input checked="" type="checkbox"/>	123456789.jpg
25	<input checked="" type="checkbox"/>	123456789.jpg
26	<input checked="" type="checkbox"/>	123456789.jpg
27	<input checked="" type="checkbox"/>	123456789.jpg
28	<input checked="" type="checkbox"/>	123456789.jpg



---

---

---

---

---

---

---



## 控制欄位屬性

動態變更caption樣式(WCUSTOMERS)

```
$('#dataFormMasterCompanyName').closest('td').prev('td').css({  
  "font-size": "200%", "background-color": "yellow", "color":  
  "#0101DF" });
```

字型大小：font-size

背景顏色：background-color

字型顏色：color



補充

## 控制欄位屬性

元件CSS控制

```
Text : $(id).css({ "font-size": "200%", "background-color": "yellow", "text-align":  
  "right" });
```

```
Refval : $(id).data("inforefval").refval.find("input.refval-text").css('background-color',  
  "#FEFFAF");
```

```
DateBox : $(id).next(".datebox").find(".combo-text").css('background-color',  
  "#FEFFAF");
```

```
ComboBox : $(id).combo('textbox').css('background-color', "#FEFFAF");
```



## 欄位計算關係

Order Details表添加一個Amount欄位，並在JQDataFormDetail上添加這個欄位  
將amount方法名稱寫在JQDataFormDetail上計算欄位的onblur屬性上

```
function amount() {  
  if ($('#dataFormDetailUnitPrice').val() == 0 || $('#dataFormDetailQuantity').val() == 0) {  
    $('#dataFormDetailAmount').attr("value", 0);  
  }  
  else if ($('#dataFormDetailUnitPrice').val() != "" && $('#dataFormDetailQuantity').val() != "")  
  {  
    $('#dataFormDetailAmount').attr("value", $('#dataFormDetailUnitPrice').val() *  
    $('#dataFormDetailUnitPrice').val());  
  }  
}
```



## 自定義Validate

(WCUSTOMERS)

在JQDataForm的OnApply屬性輸入方法名稱checkAddress

```
function checkAddress()
{
    var value = $('#dataFormMasterAddress').val();
    var str = value.substr(0,3);
    if (str == "台北市" || str == "新北市")
        return true;//可以Apply
    else{
        alert("輸入地址不符合條件!");
        return false;//不可以Apply
    }
}
```



## 加入固定文字

JQDataForm欄位後方加入文字，寫在form的onloadsuccess屬性裡(WCUSTOMERS)

```
if ($('#t1').length == 0) {
    $('#<span id="t1">最大長度10字元
</span>').insertAfter($('#dataFormMasterFax'));
}
```



## 單獨使用DataForm

☛ 對JQDataGrid隱藏。讓User表面上看起來只有DataForm (WDataFormOnly)

1. DataForm屬性AlwaysReadOnly=True。
2. JQDialog的EditMode設為 Continu模式。
3. DataForm上ToolItems屬性設置

```
<JQTools:JQToolItem Icon="icon-previous" OnClick="previous" Text="上筆"/>
<JQTools:JQToolItem Icon="icon-next" OnClick="next" Text="下筆"/>
<JQTools:JQToolItem Icon="icon-add" OnClick="insert" Text="新增"/>
<JQTools:JQToolItem Icon="icon-edit" OnClick="update" Text="更改"/>
<JQTools:JQToolItem Icon="icon-remove" OnClick="remove" Text="刪除"/>
<JQTools:JQToolItem Icon="icon-search" OnClick="query" Text="查詢"/>
```



## 使用多個DataForm

- ☛ 用來處理JQDialog中可以存放多個DataForm一起存檔(WORDERS)

JQDataForm的ChainDataFormID屬性，可以鏈接第2個DataForm一起作業(包括Load/Update/Insert)，第2個可以用CahainDataFormID指定第3個，以此類推

可以搭配JQTab元件使用



---

---

---

---

---

---

---

## 自動開啟JQDataForm

一開啟表單即可進入DataForm編輯模式的新增/修改/查詢狀態(WCUSTOMERS)

```
$(document).ready(function () {  
    $("#dataGridView").datagrid({  
        onLoadSuccess: function (data) {  
            $("#dataGridView").datagrid("selectRow", 0);  
            openForm('#JQDialog1',  
            $("#dataGridView").datagrid('getSelected'), "viewed", 'dialog');  
            //最後兩個參數分別是打開的時候狀態(viewed,  
            updated,inserted)以及JQDialog的打開模式  
        }  
    });  
});
```



---

---

---

---

---

---

---

## 元件離開焦點時的事件

(WCUSTOMERS)

1. inforefval，離開焦點時觸發Function  
 \$("#dataFormMasterCompanyName").data("inforefval").refval.find("input.refval-text").blur(function () {  
 alert("test");  
 });
2. combobox，離開焦點時觸發Function  
 \$("#dataFormMasterCompanyName").combo('textbox').blur(function () {  
 alert('combo');  
 });
3. datebox，離開焦點時觸發Function  
 \$("#dataFormMasterLastOrderedDate").datebox('textbox').blur(function () {  
 alert('datebox');  
 });



---

---

---

---

---

---

---

## DataGrid控制

INFOLIGHT

## Refval-ColumnMatch

- 在grid使用的refval通過元件的columnMatch無法做到，但是可以變通通過在OnSelect屬性上寫如下內容 (WPRODUCT)

```
function columnMatch() {  
    var refRow = $(this).data("inforefval").panel.find('table.refval-grid').datagrid('getSelected');//取得refal當前選中的資料row  
    var row = $('#dataGridMaster').datagrid('getSelected');  
    var rowIndex = $('#dataGridMaster').datagrid('getRowIndex', row);//找到被編輯的grid的rowindex  
    var editor = $('#dataGridMaster').datagrid('getEditor', { index: rowIndex, field: 'QuantityPerUnit' });//找到UnitPrice的編輯元件  
    editor.actions.setValue(editor.target, refRow.QuantityPerUnit);//給值  
}
```

INFOLIGHT

## 新增時欄位ReadOnly控制

在DataGrid 的toolItem屬性裡新增一個按鈕取代原本的新增鈕(WPRODUCT)

```
function InsertTest() {  
    var index = $('#dataGridMaster').datagrid('getRows').length;  
    insertItem("#dataGridMaster", "$(this)");//新增空白一筆記錄  
    var editIndex = undefined;  
    if (editIndex != index) {  
        $('#dataGridMaster').datagrid('selectRow', index).datagrid('beginEdit', index);  
        var cellEdit = $('#dataGridMaster').datagrid('getEditor', { index: index, field: 'QuantityPerUnit' });  
        $(cellEdit.target).attr("disabled", "disabled");//text  
        //$(cellEdit.target).combobox('disable');//combobox  
        //$(cellEdit.target).datebox({ disabled: true }); //datebox  
        //$(cellEdit.target).refval('disable');//refval  
    }  
}
```

INFOLIGHT

## 更改時欄位ReadOnly控制

在DataGrid的toolItem屬性裡新增一個按鈕取代原本的編輯鈕(WPRODUCT)

```
function UpdateTest() {  
    updateItem('#dataGridMaster');//呼叫原本的update function  
    var row = $('#dataGridMaster').datagrid('getSelected');  
    var index = $('#dataGridMaster').datagrid('getRowIndex', row);  
    var editIndex = undefined;  
    if (editIndex != index) {  
        $('#dataGridMaster').datagrid('selectRow', index).datagrid('beginEdit', index);  
        var cellEdit = $('#dataGridMaster').datagrid('getEditor', { index: index, field:  
'QuantityPerUnit' });  
        $(cellEdit.target).attr("disabled", "disabled");//text  
        //$(cellEdit.target).combobox('disable');//combobox  
        //$(cellEdit.target).datebox({ disabled: true }); //datebox  
        //$(cellEdit.target).refval('disable');//refval  
    }  
}
```



## DataGrid取得Stored Procedure資料

通過CallServerMethod的方式，返回資料，並顯示在dataGrid上，同時可以分頁顯示(WCUSTOMERS)

1. Server端的ServerMethod，返回json格式的內容

```
using System.Data;  
using Newtonsoft;  
using Newtonsoft.Json;  
  
public object[] CallMethodPaging(object[] objParam)  
{  
    string js = string.Empty;  
    //創建資料庫連接  
    IDbConnection connection = (IDbConnection)AllocateConnection(GetClientInfo(ClientInfoType.LoginDB).ToString());  
    //打開連接  
    if (connection.State != ConnectionState.Open)  
    {  
        connection.Open();  
    }  
    //開始transaction  
    IDbTransaction transaction = connection.BeginTransaction();  
    try  
    {  
        string selectSql = "select * from Customers where CustomerID like C%";  
        DataSet ds = this.ExecuteSql(selectSql, connection, transaction);  
        js = JsonConvert.SerializeObject(ds.Tables[0], Formatting.Indented);//Indented縮排 將資料轉換成Json格式  
        return new object[] { 0, js };  
    }  
}
```



## DataGrid取得Stored Procedure資料

2. 以下是分頁方法

```
function pagerFilter(data) {  
    if (typeof data.length == 'number' && typeof data.splice == 'function') {  
        data = { total: data.length, rows: data }  
    }  
    var dg = $(this);  
    var opts = dg.datagrid('options');  
    var pager = dg.datagrid('getPager');  
    pager.pagination({  
        onSelectPage: function (pageNum, pageSize) {  
            opts.pageNumber = pageNum;  
            opts.pageSize = pageSize;  
            pager.pagination('refresh', {  
                pageNumber: pageNum,  
                pageSize: pageSize  
            });  
            dg.datagrid('loadData', data);  
        }  
    });  
    if (data.originalRows) {  
        data.originalRows = (data.rows);  
    }  
    var start = (opts.pageNumber - 1) * parseInt(opts.pageSize);  
    var end = start + parseInt(opts.pageSize);  
    data.rows = (data.originalRows.slice(start, end));  
}
```



## DataGrid取得Stored Procedure資料

3. 新增一個DataGrid和按鈕，按鈕內呼叫以下JS callServerMethod的方法

```
function myServerMethod() {  
    $.ajax({  
        type: "POST",  
        url: './handler/jqDataHandle.ashx?RemoteName=sCustomers.Customers',  
        data: "mode=method&method=" + "CallMethodPaging" + "&parameters=",  
        cache: false, async: true,  
        success: function (data1) {  
            var rows = $.parseJSON(data1);//將JSON轉回到Object類型提供給Grid顯示  
            var data = new Object();  
            data.rows = rows;  
            data.total = rows.length;  
            $('#QDataGrid1').datagrid({ loadFilter: pagerFilter, url: "" }).datagrid('loadData',  
            rows);//透過loadData方法清除掉原有Grid中的舊有資料並填補新資料  
        }  
    });  
}
```



## DataGrid自動取得更新

- DataForm編輯，新增一筆後(有自動編號)，在DataGrid撈出新增那筆。(WCUSTOMERS)

在DataForm的onApplied事件，寫JS方法：

```
function autonum1(rows) {  
    var CustomerID = "";  
    if (rows) {  
        CustomerID = rows[0].CustomerID;//取得最新編號  
    }  
    $('#dataGridView').datagrid('setWhere', "CustomerID=" + CustomerID + "");  
}
```



## JQuerySingle1使用AutoNumber

新增時AutoNumber的值立即顯示在畫面上  
(WCUSTOMERS→DataGrid上EditDialog屬性可以先拿掉)

```
datagrid添加OnInserted事件  
function OnGridInserted(rowData) {  
    var rows = $(this).datagrid('getRows');  
    for (var i = 0; i < rows.length; i++) {  
        if (rows[i]["CustomerID"] == 0) { //找到新增的那筆資料  
            //顯示新值  
            $(this).datagrid('updateRow', { index: i, row: rowData });  
        }  
    }  
}
```



## DataGridDetail處理欄位自動計算, 及將加總值傳回Master存檔

```
$(document).ready(function () {
    $('#dataGridDetail').datagrid({
        onAfterEdit:
            function (rowIndex, rowData) {
                rowData.Amount = rowData.UnitPrice * rowData.Quantity;
                $(this).datagrid('refreshRow', rowIndex); } }); //加總至Master欄位。

//DataGridDetail Amount欄位Total屬性設定sum、OnTotal屬性定義此方法
function mytotal(rowData) {
    $('#dataFormMasterTotalAmt').val(rowData.Amount);
}
//DataFormMaster的OnApply屬性定義此方法
function formApply() {
    endEdit($('#dataGridDetail'));
}
```



## DataGrid 以JS程式修改欄位值

修改DataGridDetail內欄位值(WORDERS)

```
function changeGridData() {
    var data = $('#dataGridDetail').datagrid('getData');
    for (var i = 0; i < data.rows.length; i++) {
        var index = $('#dataGridDetail').datagrid('getRowIndex', data.rows[i]);
        $('#dataGridDetail').datagrid('beginEdit', index);
        var editor = $('#dataGridDetail').datagrid('getEditor', { index: index, field:
'UnitPrice' }); //欲修改的欄位
        if (editor) {
            editor.actions.setValue(editor.target, '100'); //欄位值
        }
        $('#dataGridDetail').datagrid('endEdit', index);
    }
}
```



## Detail 資料處理

使用BufferView來處理Detail資料過多的方法：

1. 將DataGridDetail的BufferView屬性設為True
2. 設定DataGridDetail的Height屬性  
例如: 300px 做為datagrid呈現時的高度  
往下捲動時透過PageSize去分批載入資料
3. DataGridDetail 的Pagination屬性 設False

BufferView只適合用在瀏覽的時候



## 元件補充

INFOLIGHT

## JQOptions使用



- WORDERS表單修改CustomerID欄位
- Editor設定為JQOptions
- EditorOption設定RemoteName為sCustomers.Customers, ValueMember為CustomerID, DisplayMember為CompanyName
- MultiSelect與OpenDialog屬性控制

INFOLIGHT

## JQBatchMove批次新增

- 將產品表中的產品批次轉入到訂單明細中
1. Server端使用前面建置的sOrders，再加入一個InfoCommand，選擇Products這個資料表
  2. 使用Wizard產生一個以JQueryMasterDetail1為範本的JQuery Webform，命名為wOrdersBatchMove

INFOLIGHT



## JQBatchMove批次新增

3. 從工具箱中拖拉出一個JQDialog元件(名為JQDialog2)
4. 在剛剛加入的JQDialog2裡，新增兩個JQDataGrid元件，命名為dgProduct、dgProduct2，分別做為查詢用的Grid和批次新增用的Grid
5. 設定dgProduct和dgProduct2的RemoteName屬性、DataMember屬性、Columns屬性，兩個DataGrid設置方式相同
6. 設定dgProduct的MultiSelect屬性為True，MultiSelectGridID選擇dgProduct2作為勾選時要同步顯示的datagrid，QueryAutoColumn屬性設為True，可以經由查詢來對產品清單做過濾



## JQBatchMove批次新增

7. 設定dgProduct的AllowAdd、AllowDelete、AllowUpdate、AutoApply、ViewCommandVisible屬性為False，dgProduct2的AllowAdd、AllowUpdate、AutoApply、ViewCommandVisible屬性為False、AllowDelete屬性為True
8. 從工具箱中拖拉出一個JQBatchMove元件至畫面上
9. 設定JQBatchMove1的SrcDataGrid屬性(批次新增資料來源)為dgProduct2，DesDataGrid屬性(批次新增目的地)為dataGridDetail，SrcSelectAll屬性設為True，代表在dgProduct2的資料進行BatchMove時會全部加入至dataGridDetail中
10. 設定JQBatchMove1的MatchColumns屬性為ProductID欄位和UnitPrice欄位



## JQBatchMove批次新增

11. 在dataGridDetail的ToolItem屬性上新增一個按鈕，並定義一個JS方法來開啟剛剛加入的JQDialog2
- ```
function openBM () {  
    $('#JQDialog2').dialog('open');  
    $('#dgProduct').datagrid('setWhere', '1=1');  
    $('#dgProduct2').datagrid('setWhere', '1=0');  
}
```
12. 在JQDialog2的OnSubmitted事件也定義一個JS方法，在按下確定鈕時將產品項目加至訂單明細
- ```
function doBM() {  
    $('#JQBatchMove1').BatchMove('Move');  
    return true;  
}
```



## JQSchedule使用

```
CREATE TABLE [dbo].[Schedule](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Description] [varchar](20) NULL,
    [FromDate] [datetime] NULL,
    [ToDate] [datetime] NULL,
    [Weekly] [int] NULL,
    [Monthly] [int] NULL,
    [FromTime] [varchar](5) NOT NULL,
    [ToTime] [varchar](5) NOT NULL,
    [Remarks] [varchar](50) NULL
) ON [PRIMARY]
```




---

---

---

---

---

---

---

---

## JQSchedule使用

- ④ Wizard產生Server端程式sSchedule
- ④ EEPNetserver->PackageManager上掛起
- ④ 使用Wizard的JQSingle1產生一個WSCHEDULE
- ④ 貼入一個JQSchedule元件
- ④ 設定RemoteName為sSchedule.Schedule，DateField為FromDate，DateToField為ToDate，TimeFromField為FromTime，TimeToField為ToTime，TipField為Remarks，TitleField為Description.
- ④ 使用JQdataGrid來新增信息，並在日曆上顯出相關的日程表
- ④ 即時更新：\$("\${.info-schedule").schedule('load');




---

---

---

---

---

---

---

---

## JQDrillDown使用

1. 使用先前建立好的兩支Server程式：sOrders和sCustomers。並在sCustomers上InfoCommand元件-Cutomers，在CommandText中添加一個欄位，用於對Orders的SUM(TotalAmt)訂單金額做加總的欄位

```
select Customers.*(select TotalAmt from (select CustomerID,SUM(TotalAmt) as
TotalAmt from Orders group by CustomerID) as TMP where
TMP.CustomerID=Customers.CustomerID)as TotalAmt from Customers order by
TotalAmt desc
```

2. 使用先前建立好的以DataGrid呈現資料的網頁：WORDERS(訂單資料)和WCUSTOMERS(客戶資料)
3. 在WCUSTOMERS表單的設計畫面上，將JQDrillDown元件從左邊工具箱中拖入




---

---

---

---

---

---

---

---

## JQDrillDown使用

- 設定JQDrillDown元件：  
RemoteName屬性設為sOrders.Orders；  
DataMember屬性選擇Orders；  
KeyFields屬性為CustomerID，作為DrillDown資料來源的鍵值欄位；  
DrillStyle屬性設為WebForm；  
(DrillStyle分為四種：WebForm、MobileForm、RDLC、Command)  
FormName屬性選擇WORDERS表單
- 設定WCUSTOMERS表單上dataGridView元件，添加進TotalAmt(訂單總金額)欄位，設定DrillObjectID 屬性為JQDrillDown1，並設定DrillField傳入的參數為CustomerID



## JQTab使用

JQMasterDetail1 樣板(WORDERS)

- JQDialog1裡拉進一個JQTab元件，以及三個JQTabItem元件
- JQTabItem 1放Master的主要欄位(使用JQDataForm)  
PreLoad=False, DataForm的ChainDataForm設定為dfMaster1
- JQTabItem2放Master的次要欄位  
(使用另一個JQDataForm,命名為dfMaster1), PreLoad=False
- JQTabItem3放Detail  
(使用JQDataGrid), PreLoad=False

事件控制: 在dataGridView的onUpdate、OnInsert屬性中設定“onInsertUpdate”事件,在JQTab1的onSelect屬性中設定“onTabSelect”事件，如下的程式:



## JQTab使用

```
var orderid = 0;
function onInsertUpdate () {
    var tab = $('#JQTab1').tabs('getTab',0);
    if (!tab.data('loaded')) {
        initControls(tab);
        tab.data('loaded', true);
    }
    /*var tab = $('#JQTab1').tabs('getTab', 1);
    if (!tab.data('loaded')) {
        initControls(tab);
        tab.data('loaded', true);
    }
    */
    $('#JQTab1').tabs('select',0); //切回Tab第1頁
    $('#dataGridDetail').datagrid({ url: '' }); //清空Detail數據
    $('#dataGridDetail').datagrid('loadData', { total: 0, rows: [] });
    orderid = 0; //清空Orderid旗標
```



## JQTab使用

```
function onTabSelect(title, index) {
    if (index > 0) { //切換頁籤時
        var tab = $(this).tabs('getTab', index);
        if (!tab.data('loaded')) {
            initControls(tab); //初始Tab內的組件
            tab.data('loaded', true); //載入數據
        }
    }
    if (index == 2) { //切換到 Detail 這頁
        var tableName = getInfolightOption($("#dataFormMaster")).tableName;
        var parentRow = getFormRow($("#dataFormMaster")); //取得Master的數據
        if (orderid != parentRow.OrderID) { //判斷是否重新載入數據
            initInfoDataGrid($("#dataGridDetail", tableName, parentRow); //載入Detail數據
            orderid = parentRow.OrderID;
        }
    }
}
```



## JQMetro使用

1. 透過EEPManager的『Menu Utility』建立MetroRoot(三層結構)
2. JQWebClient下的目錄資料夾，右鍵滑鼠Add->AddNewItem，增加一個bMetro的網頁
3. 從JQClientTools工具箱中拖入一個JQScriptManager元件，設定UseMetro屬性為True
4. 再從JQClientTools工具箱中拖入一個JQMetro元件，設定連接的DBAlias以及Metro的RootValue
5. 以DataGrid為例：從左邊JQClientTools工具箱中拖入JQDataGrid元件，設定RemoteName和DataMember屬性(與平時設計相同)
6. 點擊JQMetro元件右上角的『Edit』選項，打開設計畫面，在客戶管理的節點上，設定Size屬性為widepeek，BG Color屬性為orange，Style屬性為DataGrid，StyleObject屬性選擇剛剛拉進來的JQDataGrid1



## JQMetro使用

### JQRotator

1. 從工具箱JQClientTools下拖拉出JQRotator元件到網頁畫面中
2. 設定屬性
  - (1) RemoteName和TableName決定輪播數據的來源
  - (2) FieldName決定輪播數據來源於哪個欄位
  - (3) RotatorType決定輪播類型，分為text和image兩種
  - (4) ImageFolder是當RotatorType為image時指定圖檔來源相對於網站根節點子資料夾的相對路徑，比如圖片存放在網站目錄下的Files檔案夾中，那麼請在這個屬性中設定Files (此設定同上傳下載設定一致)
  - (5) OnBeforeLoad是在輪播開始之前的事件，參數是options，除了以上所有的屬性在options中 (首字母小寫) 之外還有額外的屬性whereString屬性可以對數據源過濾，屬性都可以通過 (參數名.屬性名) 直接修改
  - (6) MenuID設定表單在EEPManager上的MenuID



## JQMetro使用

### JQDashboard

1. 從工具箱JQChartTools下拖出JQDashboard元件到畫面中
2. JQScriptManager的UseChartJS請設定為True
3. 設定JQDashboard屬性
  - 1) Label：文字說明
  - 2) LabelColor：文字顏色
  - 3) LabelHeightAdjust：文字對於下基準線的上下偏移量可以設定正負值
  - 4) Ticks：尺規刻度值的集合
  - 5) TickColor：尺規文字顏色
  - 6) Background：儀錶板背景色
  - 7) Intervals：刻度盤中間分割劃分的界限點集合
  - 8) IntervalColors：根據Intervals區分間的界線點繪畫顏色
  - 9) RingColor：邊框顏色
  - 10) RingWidth：邊框厚度，預設是空（null）
  - 11) Value：指針對應的值
  - 12) RenderObjectID：儀錶板要render到的對象
  - 13) Width，height：儀錶板寬高
  - 14) OnBeforeLoad：顯示前執行的事件，有參數options，以上所有屬性都可以修改



## JQMetro使用

數位儀錶板的應用：透過計時器，每隔一段時間動態從後端更新最新數值到前端，不需要使用者重複的重整網頁，就能取得最新資訊

//頁面載入後，啟動設定儀錶板function

```
$(function () {  
    set_dashboard('JQDashboard1');  
});
```

//使用js標準計時器功能，每隔3000ms自動觸發更新function

```
function set_dashboard(item_id) {  
    setInterval(function () { changeValue(item_id); }, 3000);  
}
```

//每隔3000ms，此function會被觸發，去取得後端數值，並設定到儀錶板

```
function changeValue(item_id) {  
    value = 可透過程式或是serverMethod取得後端值;  
    $('#'+ item_id).plotDashBoard('setValue', value);  
}
```



## JQMetro使用

效果：每隔三秒，儀錶板數字自動更新，若超過警示數值，則以紅色顯示



效能負載



效能負載



## 其他功能

INFOLIGHT

---

---

---

---

---

---

---

## 資料Copy功能

在JQDataGrid上面ToolItem屬性添加一個新的item，並在onclick屬性中寫入call的js方法名稱(wOrdersBatchMove)

```
function copy() {  
    var row = $('#dataGridMaster').datagrid('getSelected');  
    row.OrderID = 10;//修改Key欄位  
    $('#dataGridMaster').datagrid('appendRow', row);  
    $('#dataGridMaster').datagrid('changeState', 'editing');  
}
```

或是可以直接在ToolItem的onclick屬性中呼叫copyItem方法  
Detail：copyDetailItem

INFOLIGHT

---

---

---

---

---

---

---

## 三階設計

- ☛ 修改sOrders建立第三階Table：OrderLot表
- ☛ 使用Wizard的JQueryMasterDetail2產生WORDERS2
- ☛ 從toolbox中添加一個JQDataGrid放到JQDialog1中，將其RemoteName設定對應的sOrders.OrderDetails，tableName設定為OrderLot
- ☛ 對JQDataGrid Refresh schema
- ☛ 設定JQDataGrid的ParentObjectID=JQdataGridDetail
- ☛ 設定JQDataGrid的RelationColumns(OrderID、ProductID)

INFOLIGHT

---

---

---

---

---

---

---

## SingleSignOn整合

- 給非EEP網站Add Service Reference到EEP2012發佈的位址

加入服務參考→進階→加入Web參考→選擇JQWebClient下的SingleSignOn.asmx

- 將WebService取名為Infolight
- 給需要打開EEP轉向網頁增加的Infolight加入引用using Infolight
- 增加一個按鈕，藉此SSO到JQ網站



## SingleSignOn整合

- 在Button的Click事件中，增加如下程式

```
protected void Button1_Click(object sender, EventArgs e)
{
    Infolight.SingleSignOn s = new Infolight.SingleSignOn();
    var key = s.LogOn("001", "", "ERPS", "Solution1");
    //4個參數，依次為用戶編號，用戶密碼，資料庫別名，Solution的ID

    Response.Write("<script>window.open('http://192.168.200.98/JQWebClient/SingleSignOn.aspx?PublicKey=" + HttpUtility.UrlEncode(key) + "&RedirectUrl=MainPage_Flow.aspx', '_blank');</script>");
    //規則為前部分為發佈的路徑，加上/SingleSignOn.aspx，?RedirectUrl=後面是要開啟的網頁，Publickey=後面是要上面取得的一個認證key
}
```



## 動態輸出報表

DataGrid的toolItems屬性裡添加一個按鈕，OnClick設置function名稱，比如MyPrint，在aspx中定義一個MyPrint的JS方法(WPRODUCT)

```
function MyPrint(){
    var WhereString = $("#dataGridMaster").datagrid('getWhere');//取得用戶輸入的查詢條件
    exportReport("#dataGridMaster", "sProducts.Products", "Products", getInfolightOption($("#dataGridMaster")).reportFileName, WhereString);
    //參數依序是：datagrid的id,server的某個infocommand,infocommand名字,報表文件名字,資料過濾條件
}
```

或是可以直接在item的onclick屬性中呼叫exportReport方法，並設置ReportFileName屬性



## 主頁面替換

mainpage的主網頁開啟後不顯示原有的『首頁』內容，而是顯示自己的網頁。

1. 在JQWebClient下的MainPage.aspx(如果是WF版本則是MainPage\_Flow.aspx)，搜尋「首頁」，在div裡換成一段：

```
<iframe id="main" name="main" src="MyWebs/JQSingle1.aspx"
frameborder="0" marginheight="0" marginwidth="2"
width="100%" align="right" height="628"></iframe>
```

2. 接著到MainPage.aspx.cs，將MainImg.Visible = true;註解(共兩處)



## Help Icon

- JQDialog和JQDataGrid元件的HelpLink屬性。
- 如果有設定此屬性，會在Dialog或DataGrid的右上角，多一個''的Icon，點擊時會另開此HelpLink的視窗

ID	Name	Address	Phone	Email	Website
1	1001	1001	1001	1001	1001
2	1002	1002	1002	1002	1002
3	1003	1003	1003	1003	1003
4	1004	1004	1004	1004	1004
5	1005	1005	1005	1005	1005
6	1006	1006	1006	1006	1006
7	1007	1007	1007	1007	1007
8	1008	1008	1008	1008	1008
9	1009	1009	1009	1009	1009
10	1010	1010	1010	1010	1010



## 其他

- ◎ 頁面如何讀取Menu的Items設定  
var param = Request.QueryStringByName("設定的Param名稱");  
EEP2012-SP5-CTP3版本以後請使用，取得非加密字串  
var param = Request.QueryStringByName2("設定的Param名稱");
- ◎ 主畫面中如何在頁籤中打開其他網頁。  
self.parent.addTab('Title', 'MyPage/bOrders.aspx');
- ◎ Script中,如何設計進度條  
\$.messenger.progress({ msg: 'saving controls ...' }); //進度條開始  
...//您處理的事務  
\$.messenger.progress('close'); //進度條結束





## 教育訓練滿意度調查問卷

親愛的學員，您好：

感謝您參與本次教育訓練課程，麻煩撥冗填答本問卷，提供您的寶貴意見，作為日後我們課程整體規劃與服務改善的重要參考，謝謝！（本問卷採不記名方式）

請於課程結束前填寫完畢，並將問卷繳交給工作人員，謝謝。

### 課程整體表現滿意度？

1. 講師教學表現      ☐ 很滿意      ☐ 滿意      ☐ 普通      ☐ 不滿意      ☐ 很不滿意

原因：\_\_\_\_\_

2. 教材內容      ☐ 很滿意      ☐ 滿意      ☐ 普通      ☐ 不滿意      ☐ 很不滿意

原因：\_\_\_\_\_

3. 教學環境設備      ☐ 很滿意      ☐ 滿意      ☐ 普通      ☐ 不滿意      ☐ 很不滿意

原因：\_\_\_\_\_

4. 行政服務方面      ☐ 很滿意      ☐ 滿意      ☐ 普通      ☐ 不滿意      ☐ 很不滿意

原因：\_\_\_\_\_

5. 餐點方面      ☐ 很滿意      ☐ 滿意      ☐ 普通      ☐ 不滿意      ☐ 很不滿意

原因：\_\_\_\_\_

6. 個人學習成果      ☐ 很有幫助      ☐ 有幫助      ☐ 還好      ☐ 沒有幫助

原因：\_\_\_\_\_

整體的教學滿意度      ☐ 很滿意      ☐ 滿意      ☐ 普通      ☐ 不滿意      ☐ 很不滿意

原因：\_\_\_\_\_

其他建議：

---

---

非常感謝您的耐心填寫，訊光科技將依改善的意見與建議，  
作為日後積極改進的方向，以提供您更完善的教學服務。

