

# 程式該寫在哪邊？

## 文件版本

文件版本	文件日期	編修人員	編修說明
1.0.0	2016/4/11	Frankie	描述程式碼該寫在甚麼地方

## 底層的Form

在說明程式該寫在哪邊之前，首先我們需要先大概看一下底層的幾個 Form

- BaseForm
- DBBaseForm
- EditBaseForm

## BaseForm

我們看 BaseForm 的程式碼，可以看到下面這一行，從 XtraForm 繼承下來，並且實作了 IClientForm 的介面。

```
public partial class BaseForm : DevExpress.XtraEditors.XtraForm, AppPublic.IClientForm
```

所以從 BaseForm 開始，我們提供了系統一致的外觀與行為，並且藉由實作 IClientForm 的介面來實現 MDI Form。部分程式碼在 BaseFuncs 裡面，讓未繼承自 BaseForm 的 Form 也可以藉由呼叫 BaseFuncs 來達成一致的外觀與行為。

## InitForm

BaseForm 最重要的就是提供了 InitForm 來給後面的 Form 來繼承，所有 Form 的一些基本設定都可以寫在 InitForm 裡面，通常我們會寫在 InitForm 的程式大概有下面這些：

- 設定 LookUpEdit
- 設定 CalcField
- 設定查詢
- 打開 Table (呼叫 OpenTable)

例如下面這個範例。

```

base.InitForm();

PfoInfo.SetLookupEditPair(leQContractNo, leQContractName);
InvObject.Equity.SetLookupEditPair(leQEquityNo, leQEquityName);

CommQueries.SetQueryFunc("equ_deal_buy", sbQuery, sbClear, new QueryParamObj[]
{
    new QueryParamObj() {QueryEdit = leQContractNo, Condition = " a.pno = {0} "},
    new QueryParamObj() {QueryEdit = deQTradeDateFrom, Condition = " a.trade_date >= '{0}' "},
    new QueryParamObj() {QueryEdit = deQTradeDateTo, Condition = " a.trade_date <= '{0}' "},
    new QueryParamObj() {QueryEdit = leQEquityNo, Condition = " a.code = '{0}' "}
}, "SELECT * FROM equ_deal_buy a WHERE {0}", OpenTableBy);

deQTradeDateFrom.DateTime = DateTime.Today;
deQTradeDateTo.DateTime = DateTime.Today;

SetCalc("@ContractNo", "pno", PfoInfo.GetLookup(false));
SetCalc("@ContractName", "pno", PfoInfo.GetLookup(true));
SetCalc("@EquityNo", "code", InvObject.Equity.GetLookupFunction("code", "equno"));
SetCalc("@EquityName", "code", InvObject.Equity.GetLookupFunction("code", "cname"));
SetCalc("@BrokerName", "broker", CommObjects.Broker.GetLookupFunction("broker", "cbrief"));

OpenTable();

```

## 注意

99% override 的程式一定都要有繼承的程式碼，如下面的 *base.InitForm()*；，且 99% 都在該函數的第一行。如果不是在第一行，或是沒有繼承，請合理的懷疑自己的程式是不是有錯，除非你很確定你在寫甚麼程式。

```

protected override void InitForm()
{
    base.InitForm();
}

```

## BaseForm\_KeyDown

底層在這個函數裡面將 [Enter] 鍵轉成了 [Tab] 鍵，所以當我們按下 [Enter] 之後，會跳到下一個 control 去。

## DBBaseForm

DBBaseForm 主要做了下面這些事情：

- 重新封裝資料庫相關事件

```

public void SubscribeDBEvent()
{
    if (mDataTable == null) return;
    UnSubscribeDBEvent(); //make sure only binding once!!!!
    mDataTable.TableNewRow += new DataTableNewRowEventHandler(On_TableNewRow);
    mDataTable.RowChanged += new DataRowChangeEventHandler(On_RowChanged);
    mDataTable.RowChanging += new DataRowChangeEventHandler(On_RowChanging);
    mDataTable.RowDeleted += new DataRowChangeEventHandler(On_RowDeleted);
    mDataTable.RowDeleting += new DataRowChangeEventHandler(On_RowDeleting);
    mDataTable.ColumnChanged += new DataColumnChangeEventHandler(On_ColumnChanged);
    mDataTable.ColumnChanging += new DataColumnChangeEventHandler(On_ColumnChanging);

    mBindingSource.PositionChanged += new EventHandler(On_PositionChanged);
}

```

- 提供資料庫操作的事件，第一筆、上一筆、下一筆、最後一筆、新增、刪除、修改、儲存、取消、覆核、取消覆核。

```

public void DBFirstRecord()...
public void DBPreviousRecord()...
public void DBNextRecord()...
public void DBLastRecord()...
public virtual void DBNewRecord()...
public virtual void DBEditRecord()...
public virtual void DBLockSQL(ref string lockSQL, ref string lockMsg)...
public virtual void DBDeleteDetailsSQL(ref string delSQL, ref string delMsg)...
public virtual void DBDeleteRecord()...
public virtual void DBCancelRecord()...

```

- 提供 Form 的狀態：*dsBrowse*, *dsEdit*, *dsInsert*, *dsNone*
- 提供 SetCalc 來處理 CalcField。
- 依資料字典設定 control 的屬性。

### DoWhenFormStateChange()

當 FormState 改變的時候會呼叫這個函數，所以當我們需要在 Form 狀態改變時去設定一些 Control 的狀態時，可以繼承這個函數。

例如選擇權平倉程式 frmOptDealSell3.cs 在瀏覽狀態時，leBroker.DataSource 要是全部的期貨商，但是在新增或是修改時，只要顯示該契約的期貨商。

```
protected override void DoWhenFormStateChange()
{
    base.DoWhenFormStateChange();

    switch (FormState)
    {
        case DataState.dsBrowse:
            BrokerData = dtBroker;
            break;
        case DataState.dsEdit:
        case DataState.dsInsert:
            SetBroker();
            break;
    }

    DoUpdateDetailButtons();
}
```

#### DoPositionChanged(EventArgs e)

相較於 *DoWhenFormStateChange()* 只有在 Form 狀態改變時才被呼叫，*DoPositionChanged(EventArgs e)* 可以在每次資料指標移動時被呼叫；如果我們需要依照每一筆狀態改變一些 control 時，可以 override 這個函數。

例如定存賣出 frmTdDealSell，每一筆資料要顯示買進單號，因為如果抓全部的買進單號會太多資料，所以在資料移到該筆資料的時候才去抓買進資料來設定買進單號的 LookUpEdit。

```
protected override void DoPositionChanged(EventArgs e)
{
    base.DoPositionChanged(e);

    SetTradeNoBuy();
}
```

例如 Form 要顯示總計的資料，但是實際 table 並沒有開總計的欄位，這時候可以在 *DoPositionChanged* 時計算。(應該寫 *SetCalc* 會更好)

```
protected override void DoPositionChanged(EventArgs e)
{
    base.DoPositionChanged(e);
    CalTotalAmt();
}
```

#### DoNewRow(DataRow NewRow)

當新增資料時，用來設定預設值，例如交易日預設當天等。

例如新增時，還沒存檔，所以交易單號會是空白，先寫入【系統自動產生】。



```
protected override void DoNewRow(DataRow NewRow)
{
    base.DoNewRow(NewRow);
    NewRow.FieldByName("trade_no").asString = "系統自動產生";
}
```

## 注意

DoNewRow 主要目的是用來設定 NewRow 的值，不適合用來寫其他程式。

## DoBeforePost(ref bool CanSave)

當使用者按下儲存的按鈕之後我們可能需要判斷是否可以儲存，這時候要 override DoBeforePost，若不能儲存，將 CanSave 改成 false 後 return。

例如期貨保證金異動檔 frmFutMarginFlow 存檔前會檢查該契約該日期是否已經日結，如果已經日結就不能新增當天的保證金異動檔。

```
protected override void DoBeforePost(ref bool CanSave)
{
    CanSave = false;

    if (PNo == 0) return;

    DateTime nextDate, currentDate;
    ProcessType nextProcess, currentProcess;

    PfoInfo.GetStatus(PNo, out nextDate, out nextProcess, out currentDate, out currentProcess);

    if (DateTime.Compare((DateTime)(deDate.EditValue), currentDate) < 0)
    {
        ShowMessage("[國外股票型 2 已日結]");
        return;
    }
}
```

## DBSaveOtherSQL(ref string saveSQL)

當我們存檔時需要執行其他 SQL 時，需要繼承 DBSaveOtherSQL。

例如交易對手主檔 frmBroker 變更時，需要更新交易商帳戶的型態。(應該多判斷 fee\_type 有沒有變動會更好)

```
public override void DBSaveOtherSQL(ref string saveSQL)
{
    //base.DBSaveOtherSQL(ref saveSQL);
    //證券商
    if (leType.ValueAsInteger() == 2)
    {
        saveSQL = string.Format(@" UPDATE cmn_brokeracc SET fee_type = {1} WHERE bno = '{0}' ",
            teBroker.EditValue, leFeeType.EditValue);
    }
}
```

## DoAfterPost(DataRow CurrentRow)

在存檔後需要相對應做一些處理，但是不用在同一個 transaction 裡面時使用。如果需要在同一個 transaction 請使用 DBSaveOtherSQL。

例如基金主檔 frmCmnFund 存檔之後，要預設一筆資料，但不需要跟基金主檔在同一個 transaction。

```
protected override void DoAfterPost(DataRow CurrentRow)
{
    base.DoAfterPost(CurrentRow);

    if (FormState == SysFormBase.DataState.dsInsert)
    {
        string code = CurrentRow.FieldByName("code").asString;
        string fundhouse = CurrentRow.FieldByName("fundhouse").asString;

        string sqlIns = @"
INSERT INTO fnd_funddate (pno, code, fundhouse, iss_navdt, red_navdt, iss_settled, red_settled, monthend, loguser, logtime)
SELECT TOP 1 pno, '{2}', '{3}', iss_navdt, red_navdt, iss_settled, red_settled, monthend, '{0}', GETDATE()
FROM fnd_funddate WHERE pno = {1} ";
```

## DBEditRecord()

相較於儲存，由於底層沒有提供類似 DoBeforeEdit 的函數，所以當我們要判斷能不能刪除時，只能繼承 DBEditRecord。

例如基金配息明細 frmFundDivDetail 已經過帳後就不可以修改。

```
public override void DBEditRecord()
{
    if (CurrentRow.FieldByName("book").asBoolean)
    {
        ShowMessage("已過帳不可修改或刪除!");
        return;
    }
    string ori1 = leBankNo.Value();
    string ori2 = leAcc.Value();
    Set_Bank_Acc();
    leBankNo.EditValue = ori1;
    leAcc.EditValue = ori2;
    base.DBEditRecord();
}
```

## DBDeleteRecord()

相較於儲存，由於底層沒有提供類似 DoBeforeDelete 的函數，所以當我們要判斷能不能刪除時，只能繼承 DBDeleteRecord。

例如選擇權新倉 frmOptDealBuy，刪除前判斷是否已經日結，如果已經交易日結就不能刪除資料。(如果已經日結，應該已經覆核，應該在取消覆核的時候判斷是否已經日結即可。)

```

public override void DBDeleteRecord()
{
    int pno = CurrentRow.FieldbyName("pno").asInteger;
    DateTime tradedate = CurrentRow.FieldbyName("trade_date").asDateTime;
    string s;
    if (!PfoInfo.IsEditable(pno, tradedate, false, out s))
    {
        ShowMessage("已執行交易日結，不可刪除資料!");
        return;
    }
    base.DBDeleteRecord();
}

```

DBDeleteDetailSQL(ref string delSQL, ref string delMsg)

當需要刪除相關資料的時候可以 override 這個函數。

例如選擇權平倉檔 frmOptDealSell3 刪除時，需要一併刪除沖銷檔。

```

public override void DBDeleteDetailSQL(ref string delSQL, ref string delMsg)
{
    base.DBDeleteDetailSQL(ref delSQL, ref delMsg);
    delSQL = "DELETE opt_deal_trade WHERE pno = {0} AND tradeno_sell = '{1}'";
    delSQL = string.Format(delSQL, leContractno.ValueAsInteger(), teTradeNo.Text);
}

```

LockRecord(bool Locked)

用來覆核該筆資料，可用來判斷是否可以執行覆核/取消覆核。

例如選擇權平倉 frmOptDealSell3 已經交易日結就不可以取消覆核。

```

protected override void LockRecord(bool Locked)
{
    if (!Locked)
    {
        int pno = CurrentRow.FieldbyName("pno").asInteger;
        DateTime tradedate = CurrentRow.FieldbyName("trade_date").asDateTime;
        string s;
        if (!PfoInfo.IsEditable(pno, tradedate, false, out s))
        {
            ShowMessage("已執行交易日結，不可取消覆核!");
            return;
        }
    }
    base.LockRecord(Locked);
}

```

CurrentRow

用來取得目前的 DataRow。

ReadOnlyControls

用來設定唯獨的 control。在 InitForm 裡面設定即可，如下：

```
ReadOnlyControls.AddRange(new Control[] { teTradeno, deBaseDate, dePayDate, deDivDate, leCcy, leCcyName, leDivType, ceDivUnitAmt, ceQty});
```

EditBaseForm

EditBaseForm 除了繼承自 DBBaseForm 之外，還實作了 Iprivilege，主要是提供權限相關處理。

```
public partial class EditBaseForm : SysFormBase.DBBaseForm, IPrivilege
```

基本上 EditBaseForm 主要是將 DBBaseForm 所提供的事件加以視覺化，例如提供新增、刪除、修改的按鈕等等。按鈕的 ItemClick 也是去呼叫 DBBaseForm 裡面的 DBXxxxx()。

```
private void btnFirst_ItemClick(object sender, DevExpress.XtraBars.ItemClickEventArgs e)
{
    DBFirstRecord();
}

private void btnPiror_ItemClick(object sender, ItemClickEventArgs e)...
```

```
private void btnNext_ItemClick(object sender, ItemClickEventArgs e)...
```

```
private void btnLast_ItemClick(object sender, ItemClickEventArgs e)...
```

```
private void btnNew_ItemClick(object sender, ItemClickEventArgs e)
{
    DBNewRecord();
}
```

所以 EditBaseForm 會處理甚麼時候哪個按鈕可以被按；例如按下新增之後，刪除、修改的按鈕必須隱藏。



```
protected virtual void SetBtnState(DataState dstate)
{
    if (QueryMod) return;
    switch (dstate)
    {
        case DataState.dsBrowse:
            SetBtnVisiable(new object[] {
                btnFirst, btnPiror, btnNext, btnLast, btnNew, btnEdit, btnDelete, btnQuery, btnTool,
                btnPrint, btnReload, btnClose });
            break;

        case DataState.dsEdit:
            SetBtnVisiable(new object[] { btnSave, btnCancel });
            break;
    }
}
```

## 其他常用事件

### 欄位值改變時

大部分我們會寫在 control 的 EditValueChanged 事件，例如：

- leContractno\_EditValueChanged
- leBroker\_EditValueChanged

如果是要改變值，記得要判斷 IsInModify()，如下：

```
private void leContractno_EditValueChanged(object sender, EventArgs e)
{
    if (IsInModify())
    {
        int pno = leContractno.ValueAsInteger();
        if (pno == 0) { return; }
        //trade_date
        deTradeDate.DateTime = PfoInfo.DealStartDate(leContractno.ValueAsInteger());
        deTradeDate.Focus();
        leContractno.Focus();
    }
}
```