

# 如何設定 LookUpEdit

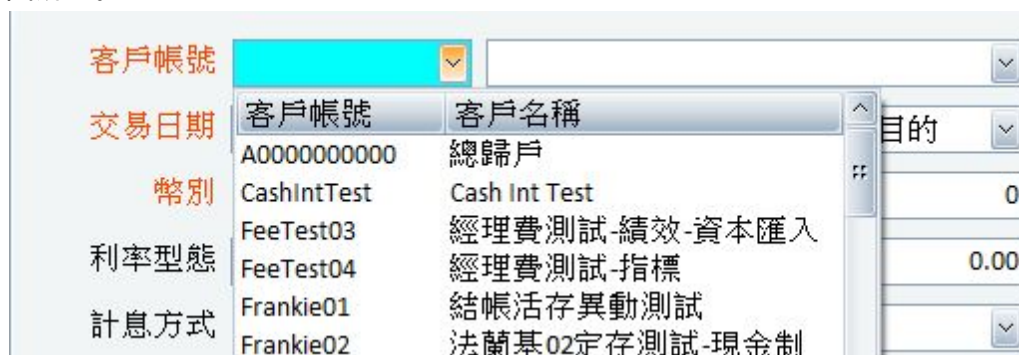
## 文件版本

文件版本	文件日期	編修人員	編修說明
1.1.0	2016/3/23	Frankie	新增【投資商品 LookUpEdit 的設定】段落
1.0.0	2016/3/21	Frankie	說明常用的 LookUpEdit 設定方式

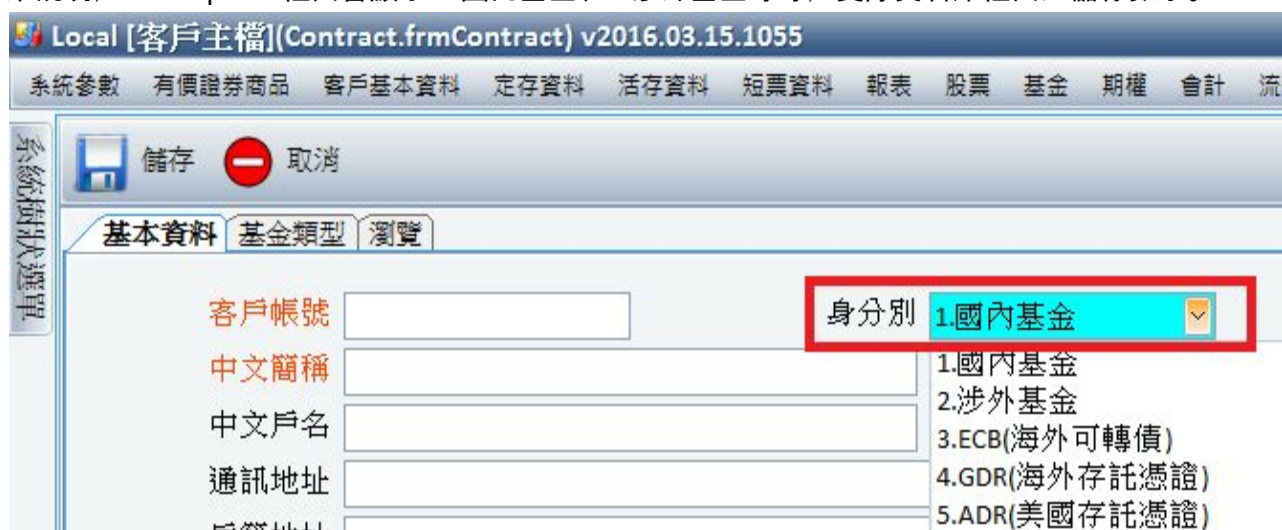
## LookUpEdit 說明

在系統的 LookUpEdit 分成兩種

1. 從基本資料取得代碼、名稱，如交易商、股票、幣別、契約等等，資料表來為儲存的是代碼。如下圖所示。



2. 資料表欄位型態為 smallint 的欄位，用來表示如 1 = 國內、2 = 國外等等；以【客戶主檔.身分別】來說明，LookUpEdit 裡面會顯示 1.國內基金、2.涉外基金等等，實際資料庫裡面只儲存數字。



在編輯的畫面中不需要特別寫程式，只需要在資料字典中設定即可。

契約主檔(contract)

中文名稱 
 選項類別 
 數值類別

預設值 
 大小寫 
☐ 唯讀

Edit Mask 
☐ 不可空值
 ☐ 新增後不可修改

欄位順序	欄位名稱	中文名稱	欄位型態	不可空值	選項類別	新增後不可修改
9	case_type	方案別	CHAR(1)	<input type="checkbox"/>	0	<input type="checkbox"/>
10	prop	身分別	SMALLINT	<input checked="" type="checkbox"/>	301	<input checked="" type="checkbox"/>

## 利用繼承自 InfoObject 的物件建立

打開 CommonUtility.CommObjects 可以看到下面這些物件均繼承自 InfoObject，所以它們全部提供了相同的方法 SetLookUpEdit 來設定 LookUpEdit。

```

namespace CommonUtility
{
    public static class CommObjects
    {
        public static InfoObject Broker;
        public static InfoObject Country;
        public static InfoObject Equity;
        public static InfoObject Future;
        public static InfoObject Issuer;
        public static InfoObject Market;
        public static InfoObject Option;
        public static InfoObject RateSource;
    }
}

```

CommonUtility.CommObjects 提供了哪些基本物件可以參考下面這邊

```
CommonUtility.CommObjects

#region 基本物件

private static CurrencyInfo curInfo;

public static CurrencyInfo Currency()...

public static InfoObject Country = new InfoObject
{
    TableName = "cmn_country",
    PKName = "country",
    NameField = "cname"
};

public static InfoObject Issuer = new InfoObject
{
    TableName = "cmn_broker",
    PKName = "broker",
    NameField = "cbrief"
}
```

打開 InfoObject 的 Source Code 我們可以看到在 InfoObject 下面有關 SetLookupEdit 的方法有這些，可以依自己的需求挑選所需要的 SetLookupEdit 來使用。

```
CommonUtility.InfoObject GetQueryValue<TResult>(string KeyField, string KeyValue, string Lookup

#region SetLookupEdit

/// <summary> ...
public virtual void SetLookupEdit(LookUpEdit LookUpControl, string displayName, string listFields, string filter = "")...

/// <summary> ...
public virtual void SetLookupEdit(LookUpEdit LookUpControl, bool isDisplayPKName, string filter = "")...

/// <summary> ...
protected void SetLookupEditPair(LookUpEdit lkCode, LookUpEdit lkName, string listFields, string filter)...

/// <summary> ...
public virtual void SetLookupEditPair(LookUpEdit lkCode, LookUpEdit lkName, string filter = "")...

/// <summary> ...
public virtual void SetLookupEditGroup(LookUpEdit[] LookUpControl, string listFields, string filter = "")...

#endregion SetLookupEdit
```

範例：

**public virtual void SetLookupEditPair(LookUpEdit lkCode, LookUpEdit lkName, string filter = "");**

設定幣別、幣別名稱

- CommObjects.Currency().SetLookupEditPair(leCcy, leCcyName);

設定交易商代碼、交易商名稱，只顯示 type = 4 的交易商

- CommObjects.Broker.SetLookupEditPair(leBroker, leBrokerName, " type=4 ");

所有 InfoObject 所提供的物件，其提供的欄位在 InfoObject 的 Constructor 裡面，必須要有提供的欄位才可以拿來設定 filter；若所需要的欄位不在提供的欄位裡面，可以討論一下是否需要放入。

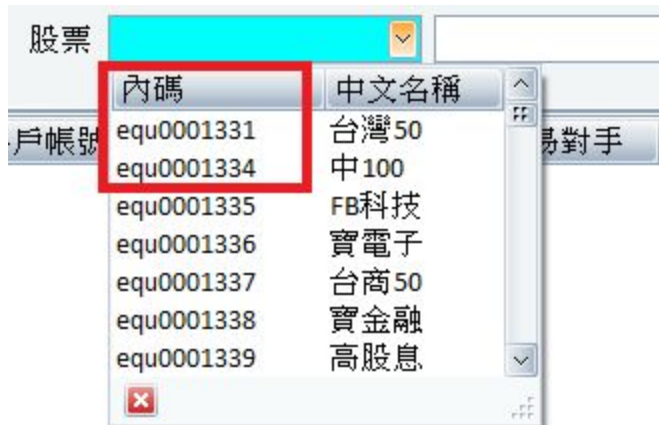
```

public InfoObject()
{
    CacheTables = new Dictionary<string, DataTable>();
    SQLTable = new Dictionary<string, string>();
    SQLTable.Add("cmn_country", "country, cname");
    SQLTable.Add("cmn_currency", "ccy, name, amtdgt, ratedgt, exchdgt, qtydgt");
    SQLTable.Add("cmn_issuer", "issuer, cbrief, cname, country, type");
    SQLTable.Add("cmn_broker", "broker, cbrief, cname, type, head, issuer, destroyed, country, is_sec, nontdcc");
    SQLTable.Add("cmn_market", "market, cname, country, type");
    SQLTable.Add("cmn_ratesource", "ratesource, sname");
    SQLTable.Add("cmn_equity", "code, equno, ticker, isincode, cname, country, ccy, cat");
    SQLTable.Add("cmn_fund", "code, fndno, isincode, cbrief, cname, fndtp, ccy, ebrief, navdgt, unitdgt, unitround, ticker");
    SQLTable.Add("cmn_bond", "code, bndno, isincode, cname, ccy, cat, country");
    SQLTable.Add("cmn_stb", "code, vouno, isincode, cat, cname, ccy");
    SQLTable.Add("cmn_future", "*");
    SQLTable.Add("cmn_option", "*");
}

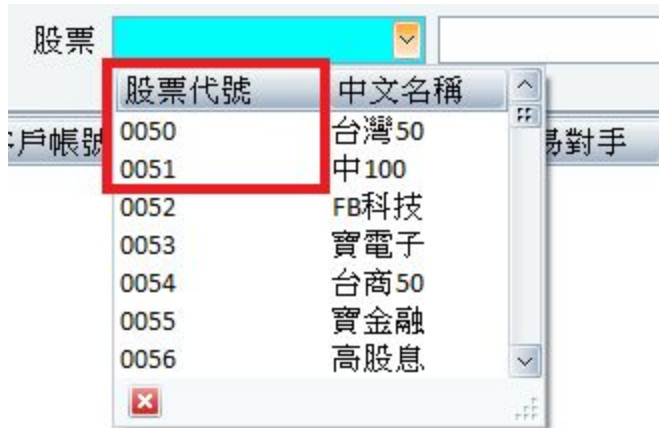
```

## 投資商品 LookUpEdit 的設定

若直接使用 `CommObjects.Equity.SetLookupEditPair(leQEquityNo, leQEquityName)`; 設定，會得到下面的結果。但是我們不是要顯示內碼，而是標的代碼。



請改使用 `InvObject.Equity.SetLookupEditPair(leQEquityNo, leQEquityName)`; 來設定 LookUpEdit。



## 設定 Contract 的 LookUpEdit

所有跟 Contract 有關的相關屬性、方法全部在 `CommonUtility.PfoInfo` 裡面，目前提供的 `SetLookupEdit` 如下圖所示。



```

public static void SetLookupEdit(LookUpEdit lk, bool isDisplayName, string filter = "")...

public static void SetLookupEditPair(LookUpEdit lk1, LookUpEdit lk2, string filter = "")...

public static void SetLookupEditGroup(LookUpEdit lk1, LookUpEdit lk2, LookUpEdit lk3, string filter = "")...

```

範例：

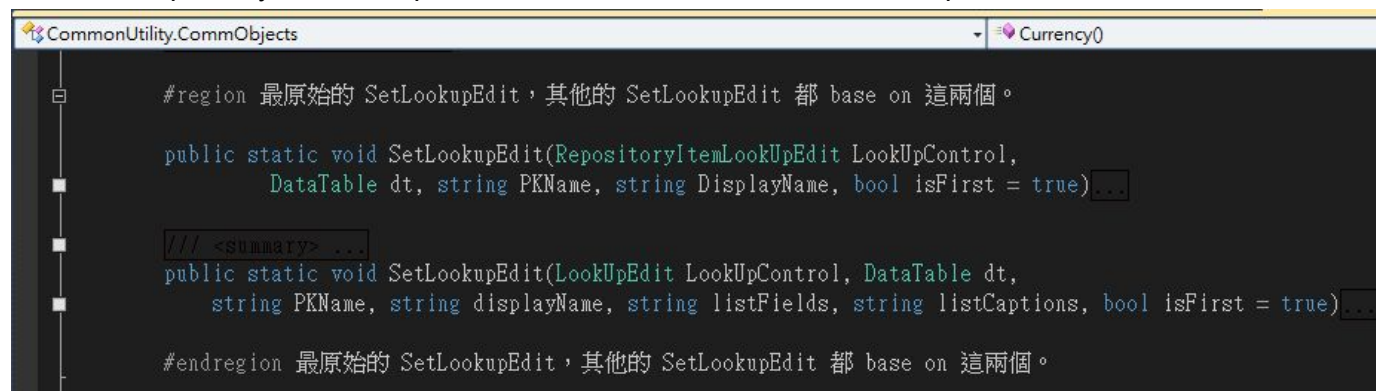
```
public static void SetLookupEditPair(LookUpEdit lk1, LookUpEdit lk2, string filter = "");
```

設定契約代碼、契約中文簡稱

- PfoInfo.SetLookupEditPair(leContractno, leCbrief);

## 最原始的 SetLookupEdit

打開 CommonUtility.CommObjects 可以找到這段程式，所有的 SetLookupEdit 基本上都是從這邊延伸出來的；其中 RepositoryItemLookUpEdit 是給 Grid 使用的，基本上與 LookUpEdit 無異。



```

#region 最原始的 SetLookupEdit, 其他的 SetLookupEdit 都 base on 這兩個。

public static void SetLookupEdit(RepositoryItemLookUpEdit LookUpControl,
    DataTable dt, string PKName, string DisplayName, bool isFirst = true)...

    /// <summary> ...
    public static void SetLookupEdit(LookUpEdit LookUpControl, DataTable dt,
        string PKName, string displayName, string listFields, string listCaptions, bool isFirst = true)...

#endregion 最原始的 SetLookupEdit, 其他的 SetLookupEdit 都 base on 這兩個。

```

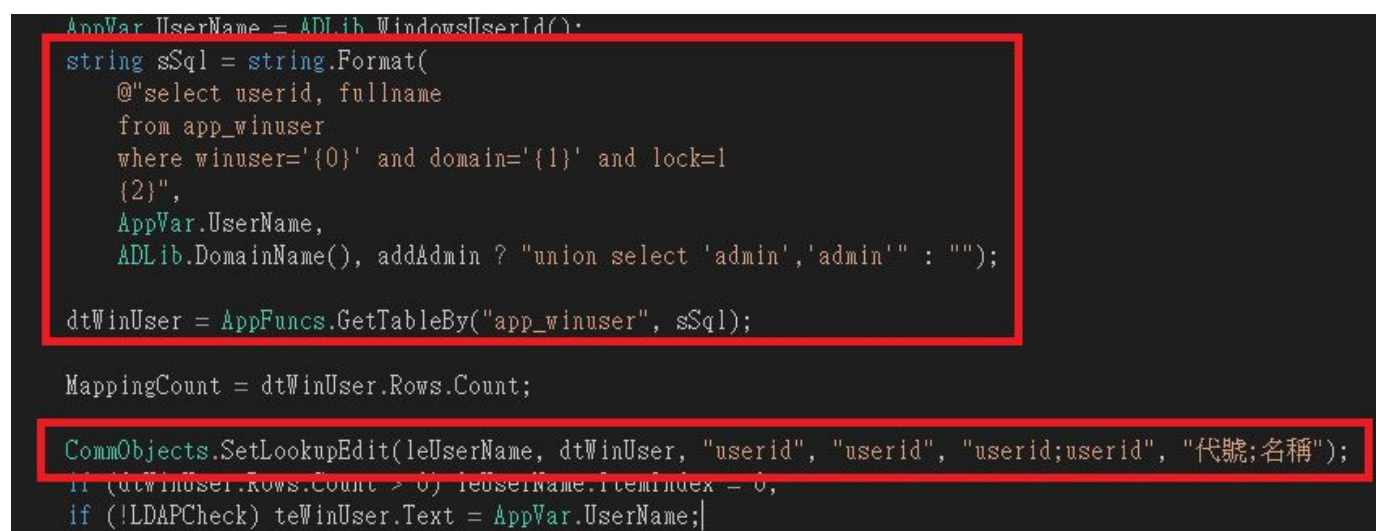
當系統提供的現成物件無符合使用的狀況可以直接使用這邊。

範例：

```
public static void SetLookupEdit(LookUpEdit LookUpControl, DataTable dt, string PKName, string
displayName, string listFields, string listCaptions, bool isFirst = true);
```

以 frmLogin 的程式來說明：

1. 取得 LookUpEdit 所需要的 DataTable
2. 呼叫 CommonUtility.CommObjects.SetLookupEdit 來設定



```

AppVar.UserName = ADLib.WindowsUserId();
string sSql = string.Format(
    @"select userid, fullname
    from app_winuser
    where winuser='{0}' and domain='{1}' and lock=1
    {2}",
    AppVar.UserName,
    ADLib.DomainName(), addAdmin ? "union select 'admin','admin' : """);

dtWinUser = AppFuncs.GetTableBy("app_winuser", sSql);

MappingCount = dtWinUser.Rows.Count;

CommObjects.SetLookupEdit(leUserName, dtWinUser, "userid", "userid", "userid;userid", "代號;名稱");
if (dtWinUser.Rows.Count > 0) leUserName.ItemIndex = 0,
if (!LDAPCheck) teWinUser.Text = AppVar.UserName;

```

所有的 SetLookupEdit 最後都會呼叫 SetLookupprop 來設定 LookupEdit 的屬性，宣告如下：

```
public static void SetLookupprop(LookUpEdit LookUpControl, bool isFirst = true)
```

在 SetLookupprop 裡面還會為 LookupEdit 掛上事件，其中以 \_lkKeyDown 跟你比較有關係，請打開 \_lkKeyDown 事件看看裡面的程式碼。

```
LookUpControl.Closed -= _lkClosed;
LookUpControl.Closed += _lkClosed;

LookUpControl.Enter -= _lkEnter;
LookUpControl.Enter += _lkEnter;

LookUpControl.KeyDown -= _lkKeyDown;
LookUpControl.KeyDown += _lkKeyDown;
```

## 注意

如果看到程式碼中有下面這行程式，請直接移除，該段程式已由底層修正。

CommonFuncs.RemoveKeyDownEvent

## 設定 Option LookupEdit

文件一開始有說明，在編輯畫面並不需要特別針對 Option 的 LookupEdit 寫程式；但是如果查詢的條件需要，還是要寫程式。SetOptionComboBox 有下面這些宣告。

```
public static void SetOptionComboBox(object c, DataTable dt);
public static void SetOptionComboBox(object c, int optIdx);
public static void SetOptionComboBox(object c, List<KeyValuePair<int, string>> list);
public static void SetOptionComboBox(object c, string tableName, string fieldName);
```

通常我們會使用第二種，如客戶契約的身分別，在資料字典裏面指定選項是 301，所以這邊我們只需要直接給 Option Index 就可以，如下所示。

- CommObjects.SetOptionComboBox(leQueryProp, 301);

欄位順序	欄位名稱	中文名稱	欄位型態	不可空值	選項類別	新增後不可修改
9	case_type	方案別	CHAR(1)	<input type="checkbox"/>	0	<input type="checkbox"/>
10	prop	身分別	SMALLINT	<input checked="" type="checkbox"/>	301	<input checked="" type="checkbox"/>

第四種的宣告其實跟第二種是一樣的，只不過他是程式自己去資料字典裡面找到選項類別；下面的寫法跟上面第二種可以得到一樣的結果。

- CommObjects.SetOptionComboBox(leQueryProp, "contract", "prop");

第三種範例如下，先宣告一個 `List<KeyValuePair<int, string>>`，然後增加需要的清單，最後呼叫 `SetOptionComboBox` 指定給 `leBankType`

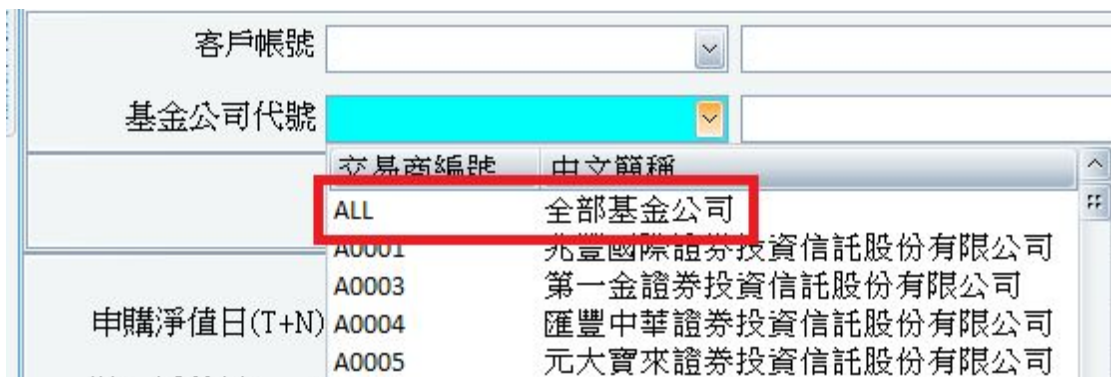
```
List<KeyValuePair<int, string>> bankType = new List<KeyValuePair<int, string>>();
bankType.Add(new KeyValuePair<int, string>(1, "1.一銀"));
bankType.Add(new KeyValuePair<int, string>(2, "2.全部"));

CommObjects.SetOptionComboBox(leBankType, bankType);
```

## 注意

目前 `List<KeyValuePair<int, string>>` 實際上應該是 `List<KeyValuePair<short, string>>`，宣告成 `int` 會造成 `EditValueChanged` 會觸發兩次；後續應該把宣告修正回 `short`。

## 增加 LookUpEdit 選項



在呼叫完 `SetLookupEdit` 之後再呼叫 `CommObjects.SetLookupEditAddAllOption`，要顯示的值自行設定，如下所示。

```
protected override void InitForm()
{
    base.InitForm();
    btnSave.Visibility = btnCancel.Visibility = BarItemVisibility.Never;
    PfoInfo.SetLookupEditPair(leQueryContractNo, leQueryChief);
    CommObjects.Broker.SetLookupEditPair(leQFndHouse, leQFndHouseName, "type=6");
    CommObjects.SetLookupEditAddAllOption(leQFndHouse, "ALL;全部基金公司");
}
```