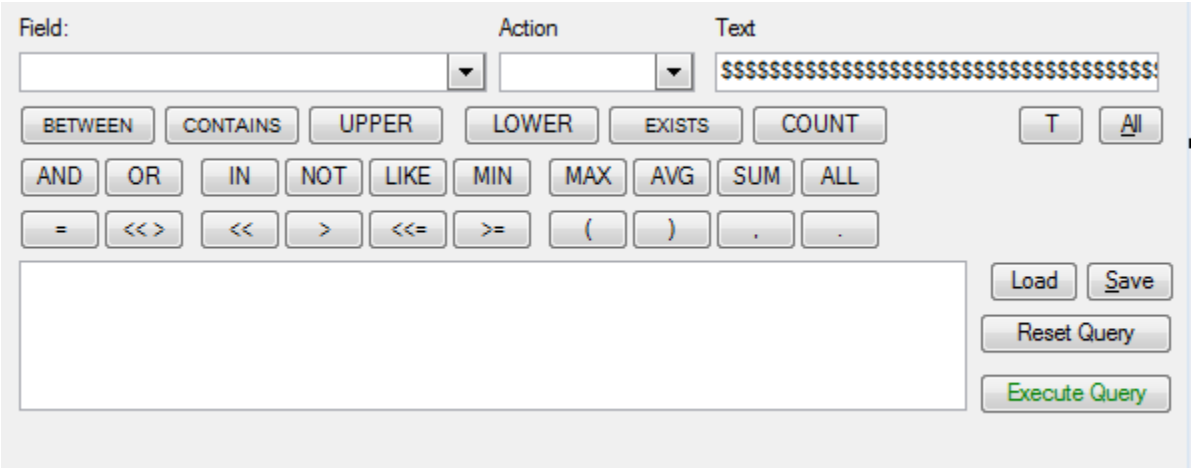
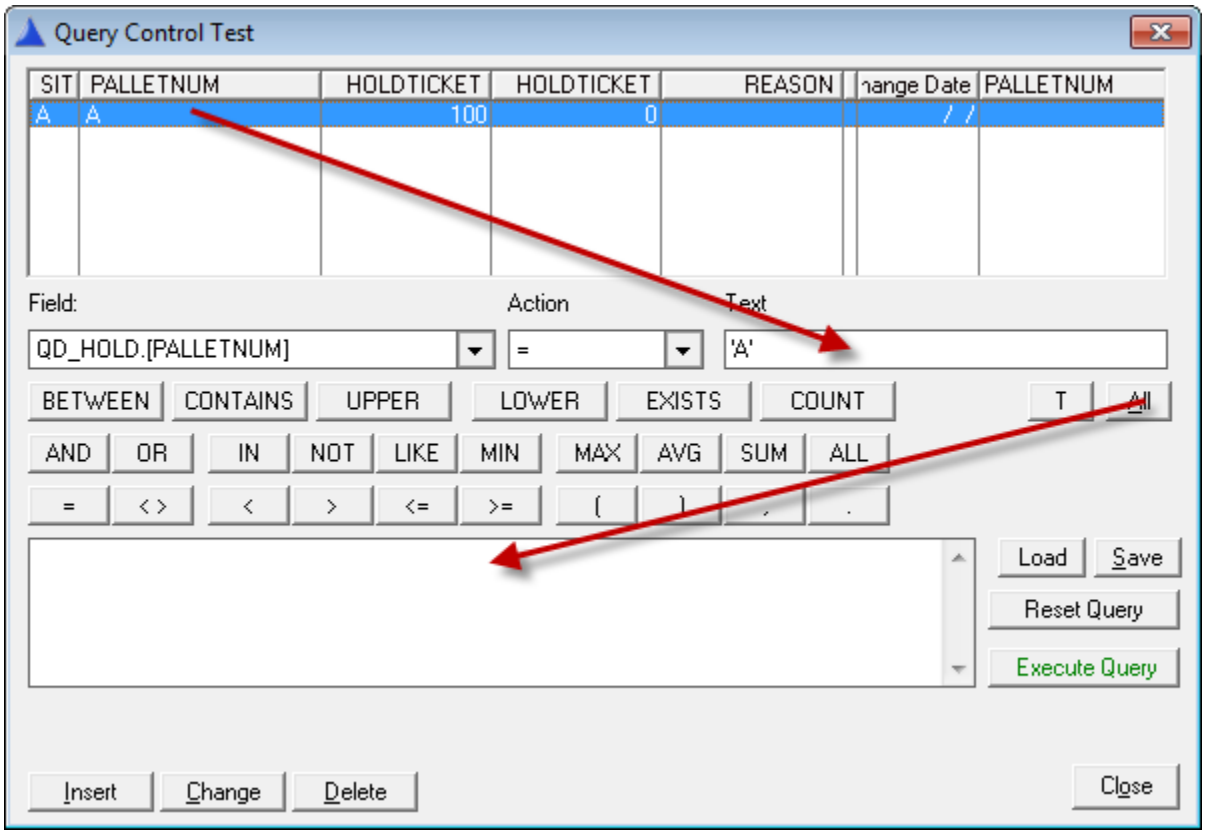
# PD CLASS GENERATOR EXAMPLE: SQL FILTER CONTROL TEMPLATE AND CLASS

## Contents

# Introduction

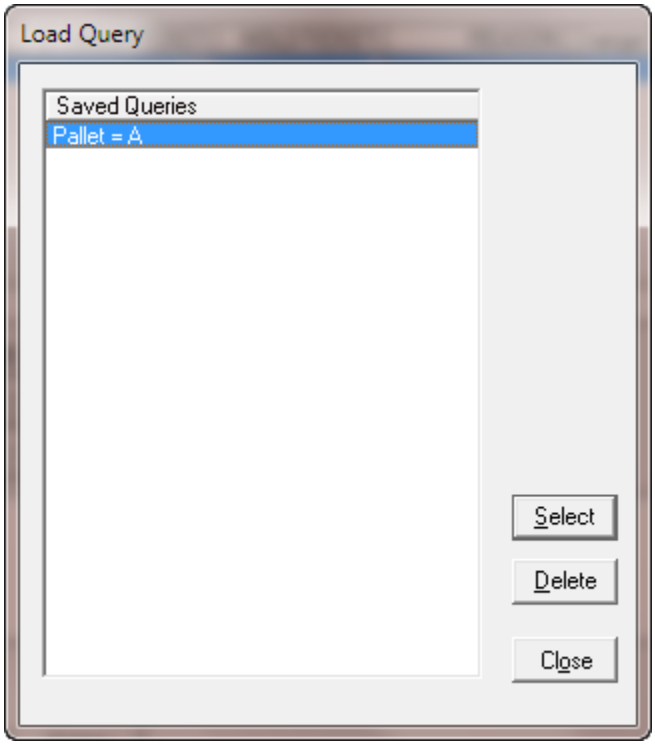


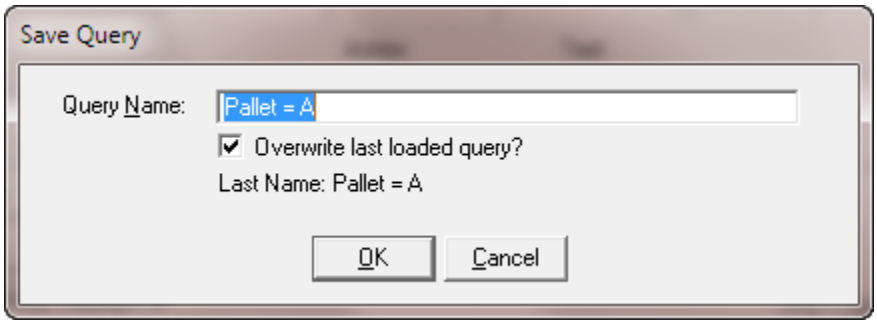

This template was derived from a Procedure developed by Robert Huff. It essentially creates and SQL Filter from fields in a view file using SQL syntax. Each time the filter is modified, it is tested to be sure that it is a valid SQL filter statement. Queries may be saved and reloaded. Fields that can be used in the query can be dragged from a listbox or from a list of file fields. The “Execute Query” button will have green or red text depending on whether the current query is valid or not. The template is designed for use with ABC applications only.



In the example show above, the entry under Palletnum in the list box can be dragged to the Text Entry, and then moved to the Query Builder field using the"All Button". This action sends the Field, Action, and Text values to the Query Builder field - the the simplest way of creating a valid Query. More complex queries can be build by typing in text , dragging fields from the Field drop list, or using the various buttons.

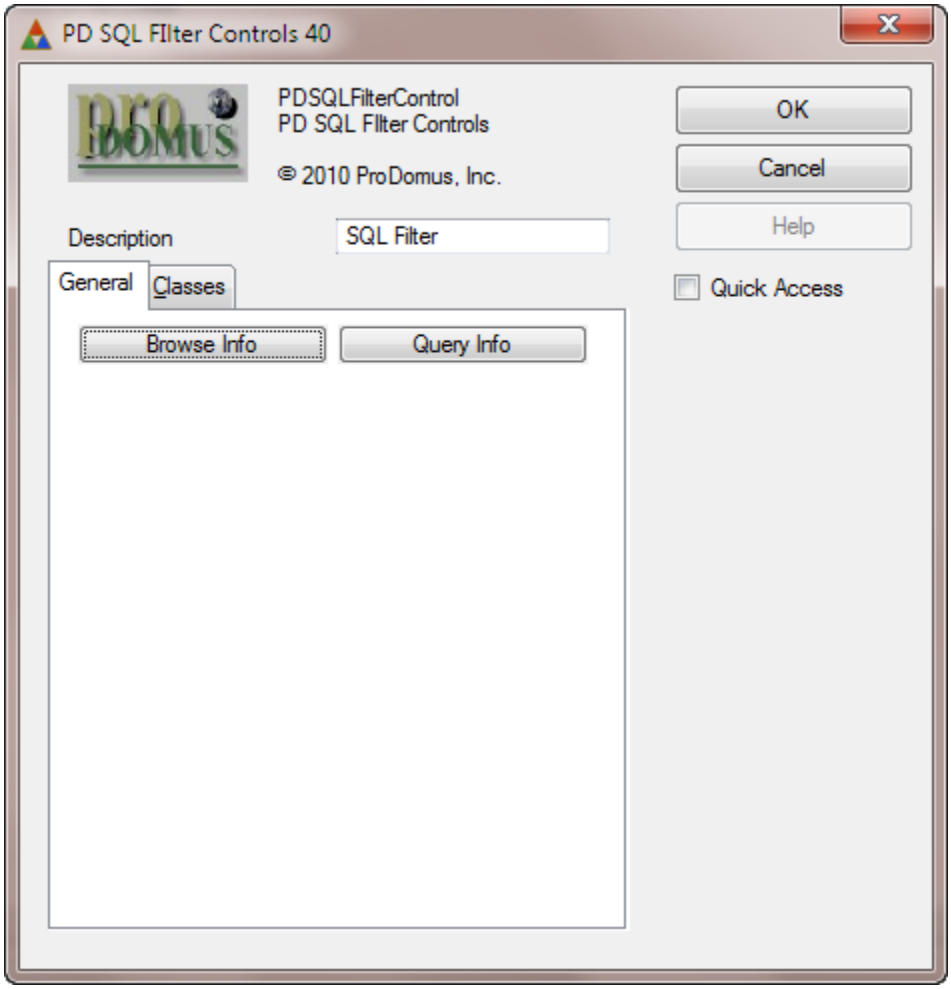
## Loading and Saving Queries



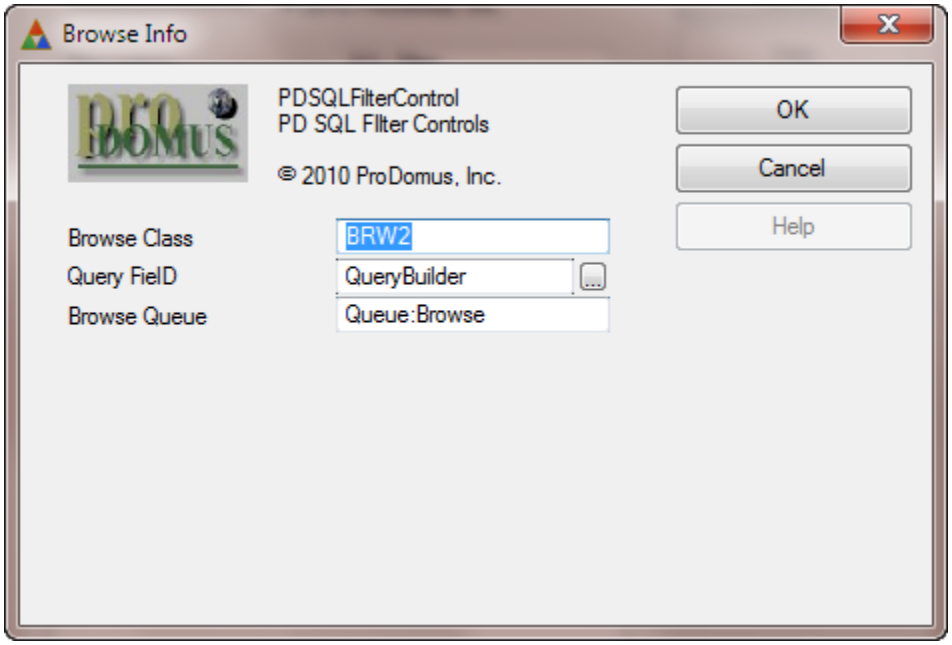

Queries may be saved and reloaded. If a previously saved query is being saved again, then the userhas an option to overwrite the existing query or save it as a different query.

## Template Entries



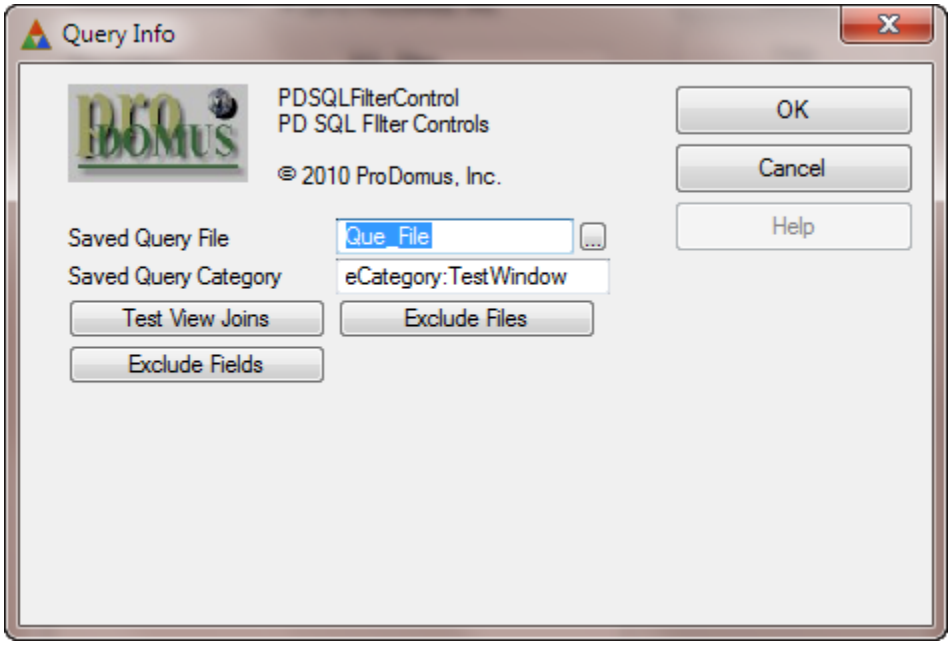

Once the controls have been populated, the template requires information of the browse box and some query related information.

## Browse Information



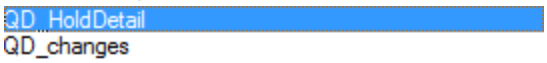

The required entries are for the Browse Class Object Name, the Query Builder Field, and the Browse Queue. The class extracts information on these for all the files and fields used.

## Query Information

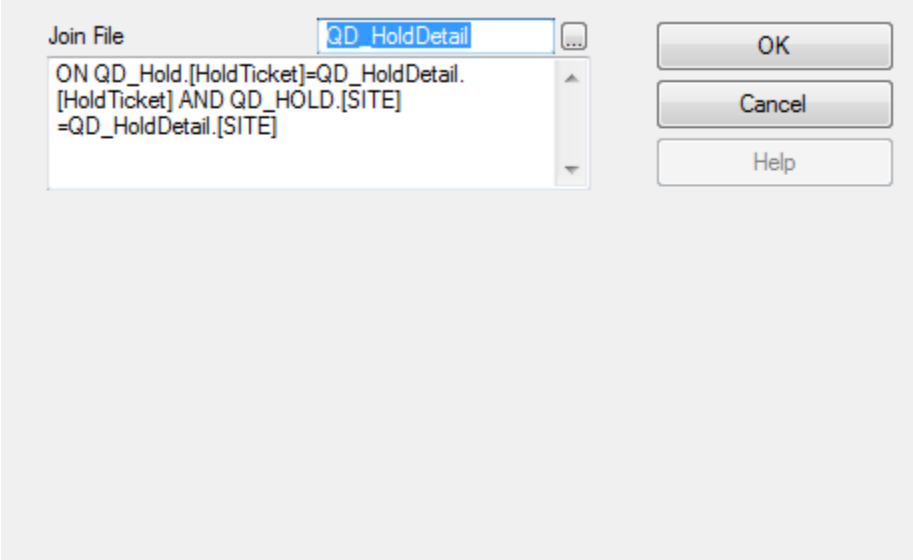



Here the Saved Query file is identified, the query category specific the browse. Any files and fields that should be excluded from queries are identified.

### Test View Joins




The class contains a view used for testing queries. This requires a list of files and their join statements using SQL syntax.  An example is shown below.




## PDSQLFilter Class Library

The Class Library, comprise of INC,  CLW, and TRN files, extracts field and file information from the browse view and query file filemanager.  By default, it expects the Save Query File to have certain characteristics. These are:

- A Primary Key with and single field.
- A secondary Key with the query category and name.

- A saved query field.

References to these are automatically assigned if the file is structured this way. If not structured this way, then several virtual methods are provided where the developer may override the default behavior.

## Query File Information, Loading, and Saving

### GetQueryFileInfo

The following properties may be assigned before the parent call. When the parent call is made, they will be assigned only if the value is NULL.

- QueryfileIDKey &KEY
- QueryFileID ANY
- QueryFileCategoryKey &KEY
- QueryfileCategory ANY
- QueryFileName ANY
- QueryFileQuery ANY

### SetQueryFile

When the list of saved queries is loaded, the QuerFileCategoryKey fields are cleared. Before the file is SET, the SetQueryFile method is called, allowing the developer to assigned values to any other fields.

### QueryFilePrimeInsert

Fields may be primed in the QueryFilePrimeInsert method before the parent call.

### QueryFilePrimeUpdate

Fields may be updated when saving a Query in the QueryfilePrimeUpdate method before the parent call.

### ValidateQueryRecord

Records may be validated when loading Saved Queries. A return value of 1 (record filtered) filters the record; 2 stops the loading (record out of range).

## Translation Methods

Two methods handle multi –language translation. If translation is turned on for an application, the template will generate Translator code to override these methods which are otherwise empty shells.

```
Translate           PROCEDURE(WINDOW pWin),PROTECTED,VIRTUAL
TranslateString     PROCEDURE(STRING pStr),STRING,PROTECTED,VIRTUAL
```

## Debug Methods

A debug method is provided for use with SysInteral's DebugView.

```
DebugOut            PROCEDURE(STRING pMsg,STRING pProc)
```

### Other Methods

Other methods most likely will not need any embed code.

```
AcceptQuery           PROCEDURE(),PROTECTED,VIRTUAL
AddExcludeField       PROCEDURE(STRING pField),VIRTUAL
AddExcludeFile        PROCEDURE(STRING pFile),VIRTUAL
AddField              PROCEDURE(SIGNED pNo,SIGNED pFEQ=0,<STRING
pAction>),PROTECTED,VIRTUAL
AddFields             PROCEDURE(*GROUP pFileMapG),PROTECTED,VIRTUAL
AddJoin               PROCEDURE(FILE pFile,STRING pJoin),VIRTUAL
AddQueryField         PROCEDURE(STRING pField),PROTECTED,VIRTUAL
AddQueryFields        PROCEDURE(),PROTECTED,VIRTUAL
BuildAction           PROCEDURE(STRING pAction),PROTECTED,VIRTUAL
BuildField            PROCEDURE(),PROTECTED,VIRTUAL
BuildText             PROCEDURE(),PROTECTED,VIRTUAL
CheckQuery            PROCEDURE(),BYTE,PROC,PROTECTED,VIRTUAL !,FINAL
DecodeQuery           PROCEDURE(STRING pCodedQuery),STRING,PROTECTED,VIRTUAL
ExecQuery             PROCEDURE(),PROTECTED,VIRTUAL
GetFieldFEQ           PROCEDURE(SIGNED pFld),BYTE,PROC,VIRTUAL
GetJoin               PROCEDURE(),STRING,PROTECTED,VIRTUAL
GetQueryActions       PROCEDURE(),PROTECTED,VIRTUAL
GoGreen               PROCEDURE(),PROTECTED,VIRTUAL
GoRed                 PROCEDURE(),PROTECTED,VIRTUAL
Init                  PROCEDURE(BrowseClass pBC,*STRING pQuery,*QUEUE
pListQueue,SavedQueryGT pQueryG),VIRTUAL
LoadQuery             PROCEDURE(),STRING,PROTECTED,VIRTUAL
LoadQueryAction       PROCEDURE(),PROTECTED,VIRTUAL
LoadQueryQ            PROCEDURE(),BYTE,PROC,PROTECTED,VIRTUAL
ParseField            PROCEDURE(*STRING pField,STRING pLabel),PROTECTED,VIRTUAL
ParseLabel            PROCEDURE(*STRING pLabel),PROTECTED,VIRTUAL
Reset                 PROCEDURE(),PROTECTED,VIRTUAL
SaveQuery             PROCEDURE(),PROTECTED,VIRTUAL
SetAlerts             PROCEDURE(),PROTECTED,VIRTUAL
SetSQLFilter          PROCEDURE(),PROTECTED,VIRTUAL
TakeAccepted          PROCEDURE(),PROTECTED,VIRTUAL
TakeDropEvent         PROCEDURE(),PROTECTED,VIRTUAL
TakeEvent             PROCEDURE(),PROTECTED,VIRTUAL
TakeNewSelection      PROCEDURE(),PROTECTED,VIRTUAL
```

## Translation File (PDSQLFilter.trn)

There is a kind of multi-purpose translation file with several sections. These are:

### Header

An itemized list of equates that can be used for filter categories. This section is referenced in the INC file.

### SourceData

Referenced in the CLW file header, this contains an equate for the field position of the Query field in the Saved Query File.  By default, this is in position 4.

It also contains a FieldMapG (Group) with a list of fields and actions associated with those fields where applicable. This group is parsed by the class library. Any modification should be done with extreme care.

## LoadQuery

This is referenced by the LoadQuery method. It contains the window whose design may be modified by the developer. Field equates and USE entries should not be modified.

## SaveQuery

This is referenced by the SaveQuery method. It contains the window whose design may be modified by the developer. Field equates and USE entries should not be modified. It also contains an equate for the text in the display of the name of the query if it has been previously save.