UT_FFUN
Create File Maintenance Functions
User's Manual

Overview

Version
07/01/2013 – 1.0

03/25/2014 – 1.1

ProIV
The ProIV Developer functions that this utility builds are designed to run on ProIV 6.1 to 7.1.

What It Does
This utility creates file maintenance functions for a selection of ProIV file definitions. The code it creates will be slightly different depending of the file type. This current version is designed to handle file types of ProISAM, SQL Server and Oracle.

A system template is available so that you can tailor the functions that are produced to your system requirements. E.g. screen size, foreground/background colours, function titles, and much more. You can also establish a unique template per file definition. This comes in handy if you want to change the default way its maintenance screen is created.

By default the screen that is created will contain a list of search parameter fields at the top with a paging screen below. All the keys of the file definition will be represented as search parameters. For the paging screen the utility will place on line 1 as many of the files keys/fields as possible until the maximum screen width is met or exceeded. The remaining data will be spread over as many lines as is necessary. These fields can be accessed by using the <F9> expand key. So you can toggle back and forth between one record per line or expanded displaying all data.

You have the ability to tailor the way a screen will be built by setting up design parameters on an individual file definition. These details are saved and all subsequent rebuilds will use these details to create the function. With tailoring you are able to add non key fields as search parameters and limit/change the order that fields will appear on the screen.

Due to ProIV limitations the maximum number of fields built into the paging screen are 255. All functions created using this utility must be genned before use.

Installation
1. Unzip UT_FFUN.ZIP into a temp folder.

2. Copy the UT.???? ProISAM files to a folder of your choice.

3. Import UT_FFUN.VPX

4. In the system I support we have different servers for Development & Production. I have placed my UT.???? ProISAM files in the Production PRODATA folder even though I only run the UT_FFUN function from within our Development environment; I do this because our Production data is backed up nightly. I have created a value variable called &$UT_FFUN_FILE_FOLDER, which I have set to "\\PFBSYN2012\D$\DATA\". Please set yours to your desired folder in step 2. UT_FFUN & UT_FFUN2 have ALIAS logic that directs the files to the appropriate folder.

5. Run UT.FFUN

Disclaimer

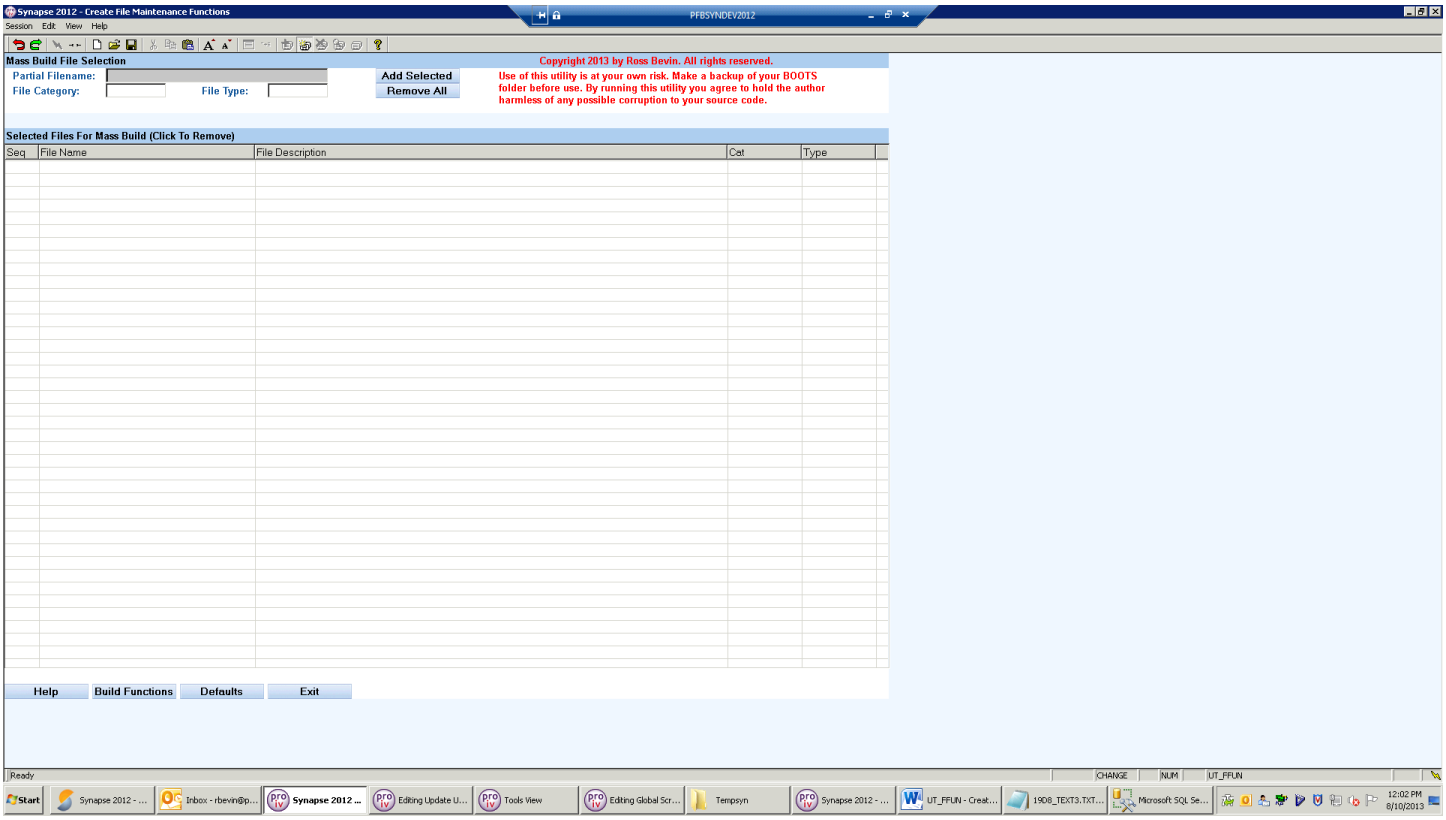
Use of this utility is at your own risk. Before you run this utility you should create a backup of your BOOTS folder. I strongly advise you do not run this function from within a production environment. It should only be run in a developments environment to be safe. The resulting maintenance functions can then be exported to your production environment. By running this utility you agree to hold the author harmless of any possible corruption to your source code.

The Author

This utility is written by me, Ross Bevin, a ProIV developer since 1985. If you have any comments or suggestions please contact me at rossbevin@yahoo.ca.

Main Screen

When you run UT_FFUN you will be presented with the following screen. This screen drives all aspects of this utility.



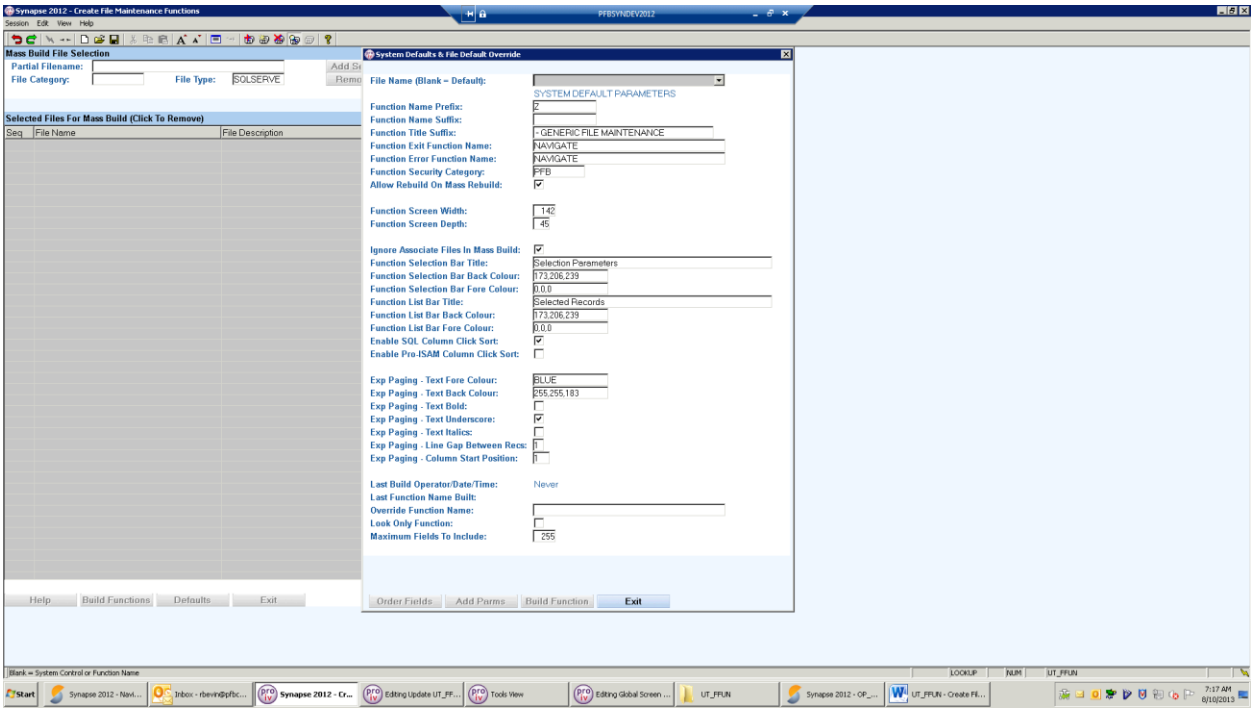
Partial Filename:/File Category:/File Type:

Enter any combination of selection criteria for the file definitions you want to build maintenance functions for. To execute the build selection list just press <enter> or click the “Add Selected” button. To clear the list completely, just click the “Remove All” button. You can also click a file definition in the list to remove it.

To build functions for all file definitions leave the above parameters blank and click the “Add Selected” button.

System Defaults

Clicking the “Defaults” button will present you with the following window.



System Settings

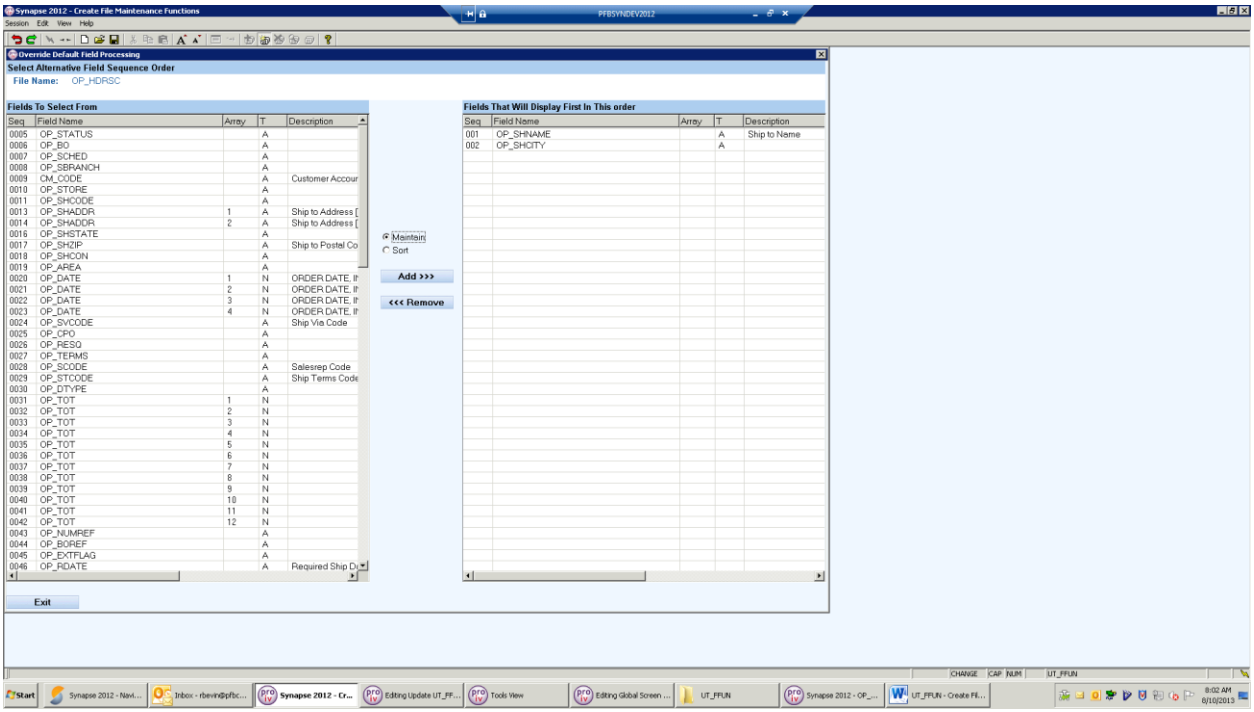
When this window opens it will display the system wide defaults that will be used to build the maintenance functions. The parameter field descriptions are self-explanatory. Whatever you change here will affect how your maintenance functions will be built.

Unique Settings By File Definition

You can establish unique settings by changing to add mode and entering the file definition name at the top of the window. As you <enter> through each field you will see that the system defaults are present. Just change the parameters you want to change for that file. When a file definition name is selected you will notice that the “Order Fields” and “Add Parms” buttons are enabled.

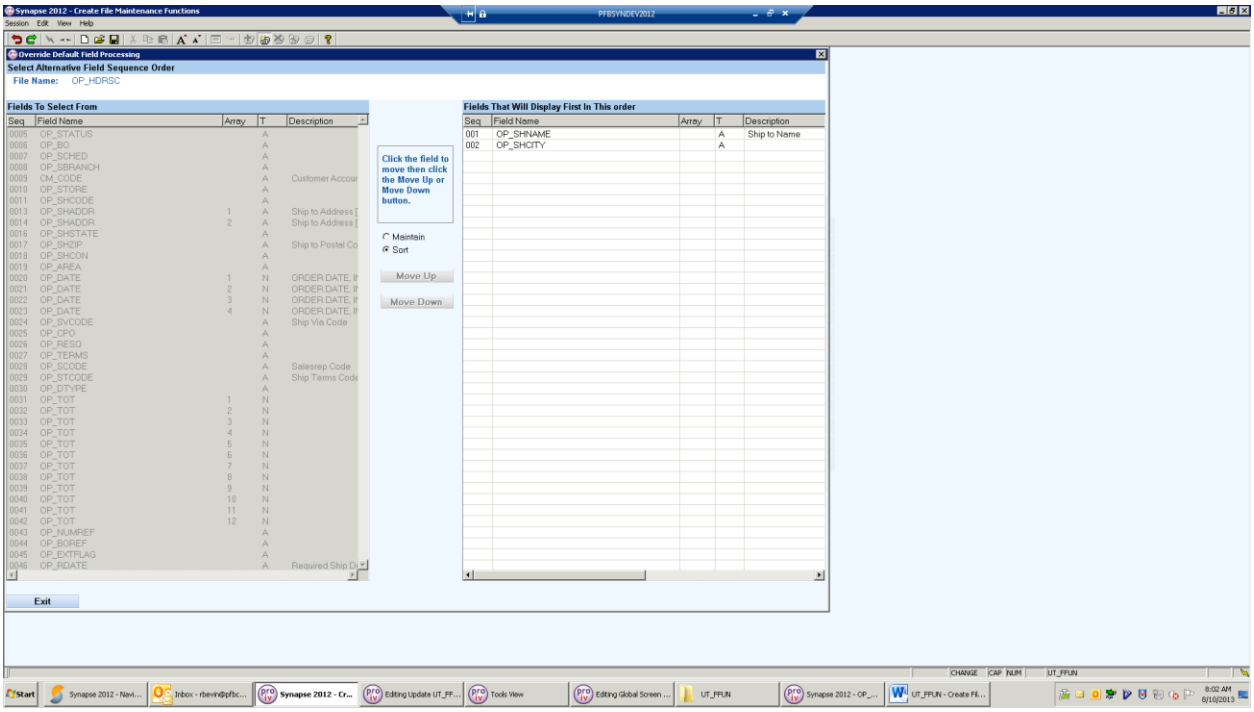
Order Fields

Clicking this button will present the following window.



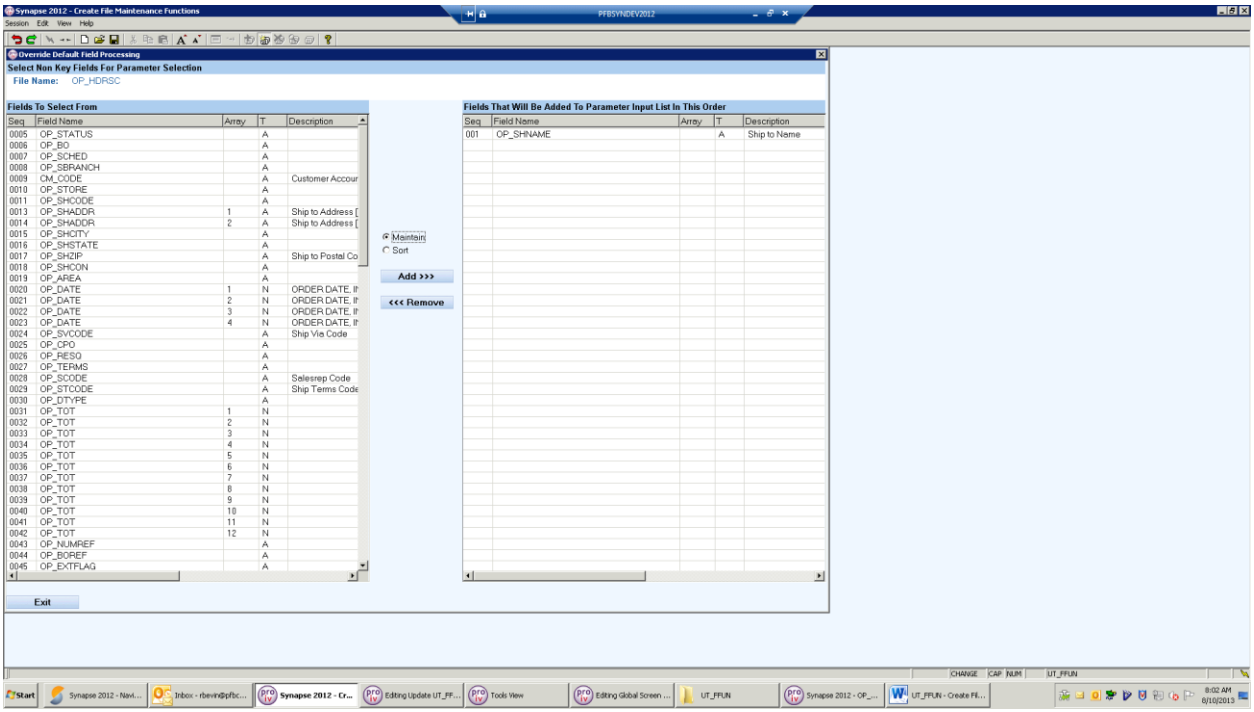
You can use this window to select those fields you want displayed first in the maintenance function. With the “Maintenance” option selected simply highlight the multiple fields you want to add or remove then click the appropriate button to execute.

Selecting the “Sort” option will change the window to the following. Use this to change the order on how the selected fields will appear in the maintenance function.



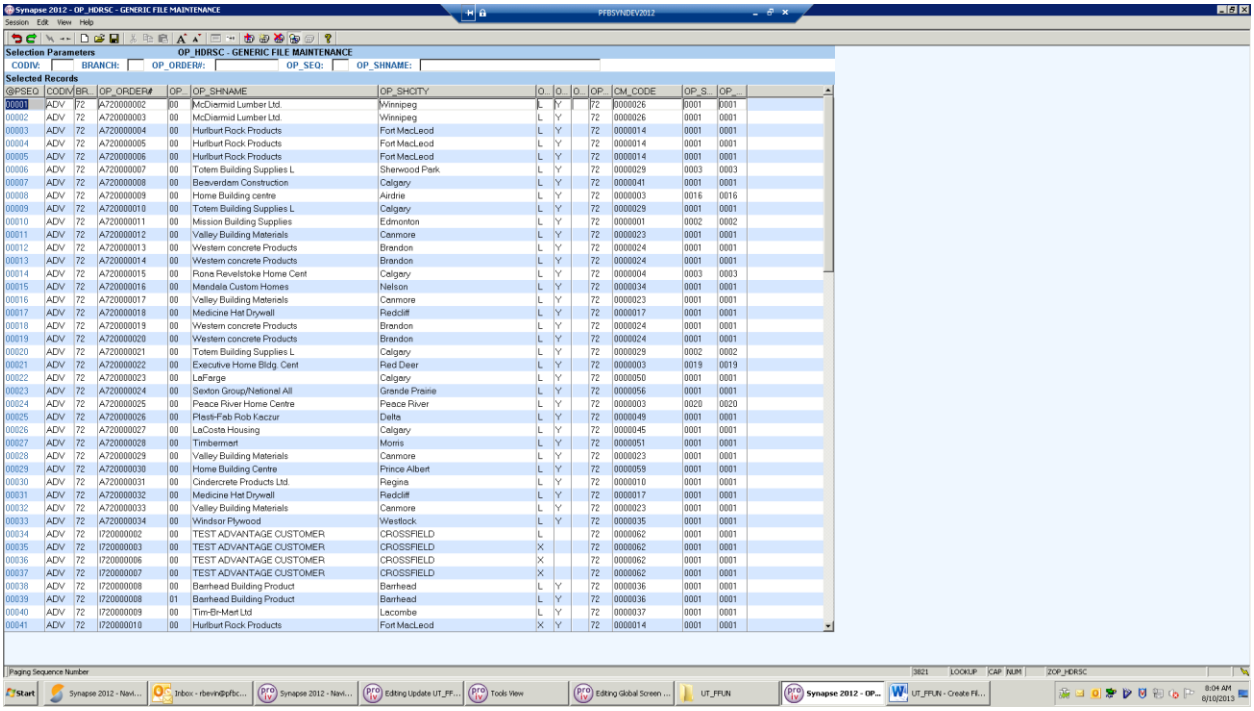
Add Parms

Clicking this button will open windows similar to those above but will allow you to select and sort fields to be added to the parameters at the top of the screen.



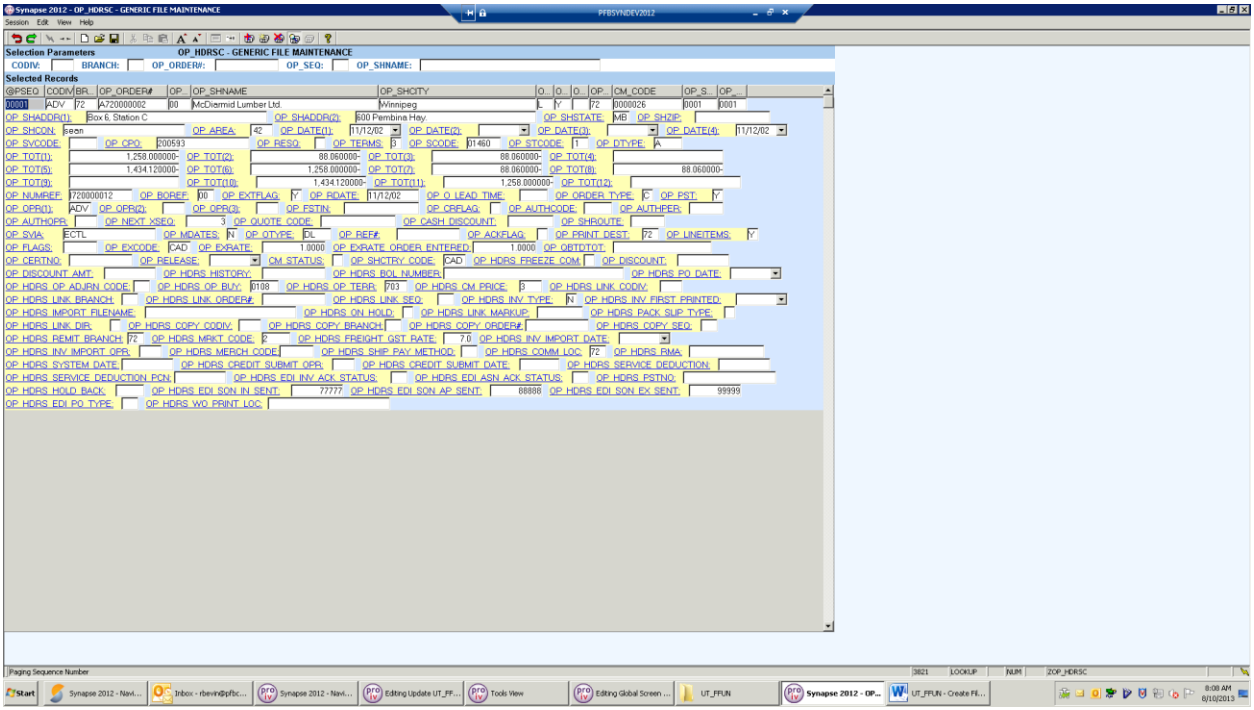
Example Function

The following example is based on the file definition (OP_HDRSC) and its unique settings as displayed in the screenshots above.



As you can see, OP_SHNAME field has been added to the parameters list at the top. Also, OP_SHNAME & OP_SHCITY are displayed first on the screen, just after the last key fields.

If you leave the parameter fields blank and <tab> then all file data will be displayed starting with the first page. When you press <F9> to expand the screen will look like this.



If you positioning the cursor on top of a field it will display that field's help description. All Julian date fields have a drop down that access a calendar function.

Parameter Wildcards

The % and _ wildcards are available when entering parameters to filter the list of records returned. Any alpha character entered is changed to uppercase and before record data is compared it is converted to uppercase too.

Following is a list of wildcard examples for selection on OP_SHNAME (store name). If the field entered ends with a '_' then the remainder of the field is padded with '_'.

W%	Returns all records with a store name starting with W.
%W	Returns all records with a store name ending with W.
%CARPET%	Returns all records with a store name containing the phrase CARPET.
W_	Returns all records with a store name starting with W.
W__T_	Returns all records with a store name that has a W in pos 1 and a T in pos 4.
W__T	The same a above but only selects on 4 character store names.
_E__T_	Returns all records with a store name that has an E in pos 2 and a T in pos 4.

The wildcards work on both ProISAM and SQLSERVE file types. A word of warning using wildcards on ProISAM files with a large amount of data. The more consecutive keys fields fully entered beginning with key 1 will result in a much faster execution. This is because a SEL-PARTIAL will be done using these keys so long as they are consecutive and they contain no wildcards. As soon as a wildcard is detected then DSEL logic takes over with filtering.